**UNIVERSIDADE DE LISBOA**

**FACULDADE DE CIÊNCIAS**

DEPARTAMENTO DE INFORMÁTICA


# TreeHop: a method to improve orthology detection

Beatriz Margarida Moço Ferreira Gomes

DISSERTAÇÃO

Mestrado em Bioinformática e Biologia Computacional

Especialização em Bioinformática

2013

**UNIVERSIDADE DE LISBOA**

**FACULDADE DE CIÊNCIAS**

DEPARTAMENTO DE INFORMÁTICA


# TreeHop: a method to improve orthology detection

Beatriz Margarida Moço Ferreira Gomes

**Dissertação orientada por:**

**José B. Pereira Leal**

Computational Genomics Lab, Instituto Gulbenkian de Ciência, Oeiras, Portugal

**Octávio S. Paulo**

Departamento de Biologia Animal, Faculdade de Ciências, Lisboa, Portugal

Mestrado em Bioinformática e Biologia Computacional

Especialização em Bioinformática

2013

# Acknowledgements

I would like to thank my supervisor José B. Pereira Leal (Zé), for taking the risk to host me as a master student. For always bringing out the best in me, for the trust and for helping me taking my first steps as a scientist.

Thank you Professor Octávio Paulo for taking care of all the bureaucracy, but especially for his initiative to create and develop this master. Thank you for giving me the opportunity to fall more in love with computers.

I thank all the people of the Computational Genomics Lab and Bioinformatics Unit for all the endless support and help. Thank you for all the laughs and the great environment. For all the criticisms that made me evolve. And for always making me feel special, one way or the other.

I would also like to thank Mónica Bettencourt-Dias for the contagious enthusiasm and for believing in me.

My master colleagues, what a great team we made! Thank you for making the geek moments less awkward and the freak-out moments more bearable!

To my former and present flatmates thank you for the awesome and unforgettable moments! "Home is not where you live, but where they understand you", thank you so much for making me feel at home every single day. Oz, Jarek, Yoan and Daisuke, you guys are the best people one could live with!

To my dear friends Andreia, Joana, Maria, Ana, Bia (which is not me) & Bota, if I could make it until here its because of you guys! Thank you for supporting all my decisions, for believing so much in me and for always being there when I needed. You guys are unique!

A special thanks to Marc Gouw. Thanks for the warm welcome the very first day. For all the precious and wise advices. The never-ending and constant help and support. For the sunny/rainy/whatever coffee breaks: made my days! Thank you also for the friendship!

I would like to deeply thank my co-worker, flatmate and friend Yoan Diekmann. Thank you for the eternal lessons, for teaching me so much! Thank you for the indescribable patience, comprehension and support. For teaching me how to think and that "one should always be very precise". For always wanting the best for me! Thanks for demanding so much and making me succeed!

To my parents thank you for the trust and respect. Thank you for making this happen, for always giving me the freedom & autonomy to choose my path without questioning. Thank you for the endless love.

To all of you, my eternal thanks!

# Resumo

Com as novas técnicas de sequenciação de genomas, a quantidade de informação a nível molecular tem crescido exponencialmente. Para perceber a origem da diversidade biológica assim como a história evolutiva de um gene, a comparação entre genomas tornou-se indispensável. Normalmente, esta comparação tem por base a análise de sequências homólogas – sequências que derivaram de um ancestral comum.

Há pelo menos dois sub-tipos de homologia: ortologia e paralogia. Genes ortólogos (ortho = exacto) são genes homólogos que derivaram de um ancestral comum através de um evento de especiação. Genes parálogos (para = paralelo) são genes homólogos que derivaram de um ancestral comum através de um evento de duplicação.

Genes ortólogos são importantes para estabelecer a correspondência entre genes de espécies diferentes; são os únicos que reflectem a árvore das espécies e por isso a reconstrução de árvores filogenéticas tem de ser baseada neste tipo de genes; na maioria dos casos, genes ortólogos têm funções equivalentes em diferentes organismos sendo por isso utilizados para a anotação de funções.

A detecção de genes ortólogos não é uma tarefa fácil devido a vários factores, entre eles: perda, duplicação, fusão e fissão de genes, e eventos de transferência horizontal. Além destes eventos biológicos, a composição das proteinas pode também afectar a detecção destes genes, como no caso de proteínas com mais do que um domínio ou com domínios de pouca complexidade (por exemplo, proteínas coiled coil).

Com o intuito de ultrapassar alguns destes obstáculos, foram criados diversos métodos para a detecção de ortólogos (até à data mais de 30). Em geral, estes podem ser divididos em duas categorias: métodos baseados na formação de grafos (graph-based) e métodos baseados em filogenia (tree-based). Os primeiros formam "clusters" de ortólogos baseados na semelhança entre pares de sequências, distinguem menos relações evolutivas mas são mais eficientes. Os segundos têm mais precisão mas requerem maior poder computacional. Para estudos em grande escala, o custo computacional poderá tornar-se um factor limitante.

Neste estudo, nós propomos um novo método a que chamámos TreeHop que tem como objectivo combinar a eficiência de métodos baseados em grafos usados em grande escala com a precisão de métodos filogenéticos usados em pequena escala.

O TreeHop foi pensado para funcionar como uma extensão de um qualquer outro método de detecção de ortólogos já existente (método base), que pode ser baseado em grafos ou em filogenia. O seu *input* é um perfil de genes ortólogos detectados pelo método base para um dado gene e uma árvore para um determinado conjunto de espécies. O TreeHop tem como objectivo inferir ortólogos nas espécies para as quais o método base não os conseguiu detectar, e às quais nos referimos como espécies gap. Para cada espécie gap, o TreeHop utiliza o ortólogo detectado na espécie mais próxima para procurar um possível ortólogo nesta espécie. Se não

o encontrar continua a percorrer a árvore da espécie mais próxima para a menos próxima, até encontrar um ortólogo ou não haver mais espécies de onde saltar.

O método usado como base e o método usado para a detecção de mais ortólogos (método de salto) são independentes: podem ser o mesmo ou diferentes. Mas é de notar que o método de salto tem de ser um método pairwise.

Nesta tese, nós utilizámos o método Bi-directional Best Hit como método base e método de salto. Este assume que genes são ortólogos se forem o primeiro hit (BLAST), reciprocamente em dois genomas. Uma das desvantagens deste método é o facto de ser apenas capaz de inferir relações um-para-um. Mesmo assim, esta continua a ser uma das metodologias mais usadas devido à sua eficácia. Também foi mostrado que a sua performance é melhor em comparação com alguns métodos mais complexos.

Devido à falta de um *gold standard* em larga escala, a validação do algoritmo foi realizada contra um método baseado em filogenias PhylomeDB, que é uma base de dados pública para colecções completas de filogenias de genes. Foi também feita uma validação, em pequena escala, contra um conjunto manualmente curado de 70 famílias de proteínas cuja composição apresenta desafios a nível biológico e técnico, à detecção de ortologia.

Foram feitas diversas análises para testar a robustez do algoritmo: analizou-se como a escolha do método base afecta a qualidade dos resultados; assim como a qualidade da árvore das espécies; testou-se também se o TreeHop tinha uma performance inferior em certas classes de proteínas (proteínas com mais de um domínio, família de proteínas de grande número e proteínas com regiões de baixa complexidade).

Tentou-se também perceber se determinados parâmetros poderiam ser mudados para melhorar a performance do TreeHop. Entre eles testou-se um método de salto diferente, o efeito do e-value do alinhamento e tamanho de proteínas, e diversas estratégias de salto.

No final, concluímos que o TreeHop aumenta a sensibilidade e precisão do método base e propomos também algumas heurísticas para modular a sua performance.

# Abstract

Reliable prediction of orthologs – genes descending from a common ancestor through a speciation event – is critical for comparative and evolutionary genomics as well as for functional annotation transfer. Phylogenetic approaches are known to be accurate, however, they are computationally expensive which becomes a limiting factor for large-scale analyses. On the other hand, graph-based methods which cluster orthologs based on pairwise sequence similarity of proteins distinguish less evolutionary relationships but are more efficient.

Here, we propose a novel orthology detection method coined TreeHop that aims to combine the efficiency of large-scale pairwise methods and the accuracy of small-scale phylogenetic approaches. TreeHop was designed to work as an extension of any other existing orthology detection method, in the following referred to as base method, and makes use of a given species tree. Based on the assumption that it is more likely to find orthologs between closely related species, TreeHop exploits the orthologs found in the closest species in order to search for more orthologs that the base method may have missed. We validated our algorithm against PhylomeDB which is a public database for complete collections of gene phylogenies and against a set of manually curated protein families composed of different technical and biological challenges for orthology detection. We find that TreeHop increases the sensitivity and accuracy of the base method and propose several heuristics to modulate its performance.

**keywords:** orthology detection, orthologs, distantly related species, large-scale analysis, evolutionary relationships

# Contents

# 1 Introduction

Evolution has generated the overwhelming diversity of proteins, cells and species we observe today. In order to understand this diversity, comparative approaches at the genetic, morphological and behavioural level are essential. The comparison between two biological entities starts with the description and identification of their corresponding parts. Central to this is the concept of homology, because "whenever we ask if two characters are the same [...], we are asking if they are homologous" (Ereshefsky, 2012).

## 1.1 What are homologs?

The term homology was first introduced by Richard Owen in 1843, as "the same organ in different animals under every variety of form and function". In a broader sense, homology can be defined as the relationship of any two characters (genic, structural or behavioural) that are derived from a common ancestral character (Fitch, 2000). In this study our focus will be on homologous genes, *i.e.* genes sharing a common origin. There is a distinction between different sub-types of homology, because at the gene level homology is not sufficient to describe the evolutionary relationships between genes.

### 1.1.1 Distinction between the different sub-types of homology

In 1970, Walter Fitch coined the terms of orthology and paralogy to distinguish two main sub-types of homology.

Orthologs (*ortho* = exact) are homologous genes that are derived from a single gene via a speciation event in the last common ancestor of the compared species (Fitch, 1970).

Figure 1.1 illustrates the evolutionary history of a gene that derived from a Last Common Ancestor (LCA) and descended to three extant populations. The gene $x$ in species $A$ ($A_x$) and gene $x$ in species $B$ ($B_x$) are an example of a pair of orthologous genes: gene x was present in the last common ancestor (LCA) of species A and B, after a speciation event $S_1$, the *same* gene was kept in both of species A and B.

Paralogs (*para* = in parallel) are defined as homologous genes that are derived from a single gene, via a duplication event (Fitch, 1970). Paralogous genes can further be sub-classified according to their time of emergence into in- and out-paralogs: if the speciation occurs before the duplication event, the duplicated genes are named *in-paralogs*; if the speciation event occurs after the duplication event, the duplicated genes are referred to as *out-paralogs* (Sonnhammer and Koonin, 2002). This classification depends on which speciation event we are referring to. For example, if we consider the speciation event $S_2$ in Figure 1.1, the genes $C_{x'}$ and $C_{x''}$ are *in-paralogs* because $S_2$ comes before $D_2$. On the other hand, $C_x$ and $Cx'$ are *out-paralogs* because the ancestral node between these two genes is the duplication event $D_1$ which comes before the speciation $S_2$. If we now consider the speciation event $S_1$, all the

duplicated genes are considered *in-paralogs* since all duplications, $D_1$ and $D_2$, occurred after $S_1$.

Co-orthology is a relation that combines the concepts of *orthology* and *in-paralogy* (Sonnhammer and Koonin, 2002). For example, the two in-paralogous genes $B_x$ and $B_{x'}$ in Figure 1.1 are co-ortholgs to the gene $A_x$ with respect to the speciation event $S_1$. Orthologs can then be sub-classified according to the number of relationships between each other, into one-to-one, one-to-many (e.g. $A_x$ and $B_xB_{x'}$) and many-to-many (e.g.$B_xB_{x'}$ and $C_xC_{x'}C_{x''}$) orthologs. Note that *many* indicates that the gene underwent a duplication event after the speciation between the two species.

Orthology and paralogy are not the only sub-types of homology. For example, xenology refers to the relationship between two homologous genes that emerged by horizontal gene transfer between two different species (Fitch, 2000). Despite their importance, xenologs are not further considered in this thesis.

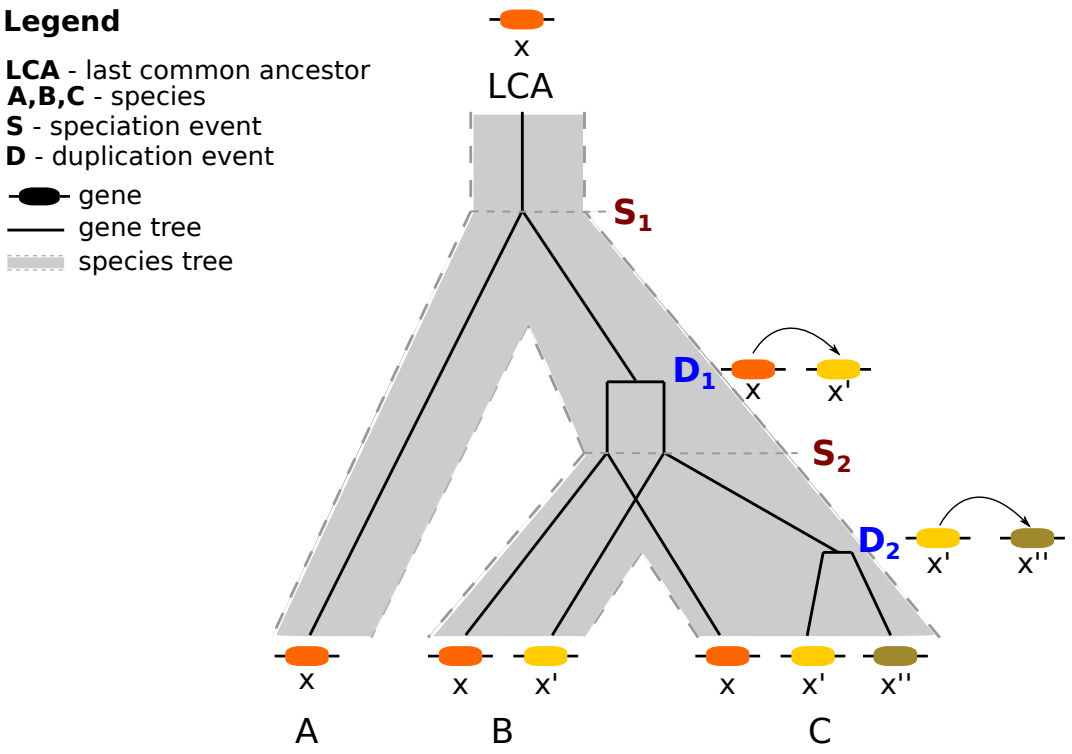



**Figure 1.1:** The evolution of gene $x$ (black line) from a common ancestor (LCA) in an ancestral population (grey background), descending to three populations(A, B and C). There are two speciation events $S_1$ and $S_2$ and two duplication events $D_1$ and $D_2$. The leaves of the gene tree (black line) represent extant genes. Genes $x'$ and $x''$ symbolize genes that emerged via duplication. All pairs of genes are homologous to each other, they all derived from a common ancestor (LCA). Genes that derived from a speciation event are orthologous genes, for example $A_x$ and $B_x$. Genes that derived from a duplication event are paralogous genes, for example $B_x$ and $B_{x'}$.

## 1.2 Why is the detection of orthology important?

Orthology is the evolutionary concept that allows us to talk about the *same* gene in different species. As seen in Figure 1.1, the orthologous genes are the only ones that reflect the species tree, making them a pre-requisite for phylogenetic analysis (Fitch, 1970). Evolutionary relationships between several species have to be based on orthologous sequences.

For biologists, orthologs gain even more importance due to the fact that they frequently have equivalent functions. Therefore, they can be used for functional annotation transfer from one organism to another, usually from a model organism to a newly sequenced genome. This transfer of functional annotation is stated in the 'orthology function conjecture': orthologs carry out biological equivalent functions; by contrast, paralogs functions typically diverge after duplication. Several studies support this theory showing that orthologs are more conserved at different levels: domain architecture and intron positions (Forslund et al., 2011); protein structure (Peterson et al., 2009); and tissue expression (Huerta-Cepas et al., 2011b). However, this conjecture has been challenged by studies claiming that paralogs within the same organism are more closely related functionally than orthologs in different organisms at the same level of divergence (Nehrt et al., 2011; Studer and Robinson-Rechavi, 2009).

## 1.3 Why is orthology inference difficult?

The amount of methods and databases already designed (more than 30 according to the Quest For Orthologs[1]) to detect orthologs, suggests that this problem has not yet been solved. Indeed, some events such as gene fusions/fissions, gain and loss of protein domains, gene loss as well as horizontal gene transfer contribute to the appearance of false or miss predictions (Koonin, 2005).

Gene fusions and fissions as well as gain and loss of protein domains can lead to multi-domain / hybrid proteins. This can complicate orthology assignment. For example, in one species a gene coding for a multi-domain protein is orthologous to two or more genes coding for each of the domains in other species. This means that a multi-domain / hybrid protein can contain domains that do not share a common ancestor. This gives rise to a conceptual problem: should we talk about orthologs at the gene or domain level? The original definition of orthology is genocentric, however, these biological events require a change in the evolutionary unit from gene to domain (Kuzniar et al., 2008; Koonin, 2005).

Gene loss and gene duplication can also lead to false positive orthology predictions. For example, in case of an ancestral duplication followed by a gene loss in one of the compared genomes can lead to one-to-one orthology assignments when actually the genes are out-paralogs. This means that out-paralogs can be inferred as orthologs when true orthologs are loss (Scannell et al., 2006).

[1] `http://questfororthologs.org/orthology_databases`

Horizontal gene transfer (HGT) can lead to erroneous interpretations. For example, when comparing the same gene from two different genomes in which one of them was acquired by horizontal gene transfer, they should be called xenologs, although such a pair of genes would mimic orthologs. Despite the fact that they can refer to the same ancestral gene, these genes do not fit the definition of orthology, *i.e.* a pair of genes that derived via a speciation event (Koonin, 2005).

Not only biological events can lead to challenging scenarios for orthology detection, but also some protein compositional biases may affect this task. For example, coiled-coil domains which consist of two to five $\alpha$ -helices that twist around one another to form a supercoil, share regions of low complexity called heptad repeats. These regions are structurally constraint and diverge less, hence are more likely to be similar by chance even if they are not homologs (Rackham et al., 2010; Coletta et al., 2010).

## 1.4 How can we infer orthology?

Usually, high sequence similarity is interpreted as evidence for homology. But this implication is not always true: two proteins can be similar because they diverged from a common ancestor, or because they converged from different ancestors, referred to as analogous proteins. The question of how their similarity arose could be answered if we had access to the ancestral sequences (Fitch, 1970). Since ancestral sequences are in most cases impossible to obtain, a variety of computational methods have been developed to infer evolutionary relationships between sequences in extant species. These methods can be grouped into two main classes: graph-based and tree-based methods.

### 1.4.1 Graph-based methods

Graph-based methods rely on pairwise comparisons between pairs of genomes. They usually comprise two phases: graph construction phase in which pairs of genes (nodes) are inferred and connected by edges and a clustering phase in which groups of orthologous genes are constructed. An orthologous group is the sum of genes and evolutionary relationships: orthology, in-paralogy and co-orthology (Kuzniar et al., 2008; Kristensen et al., 2011).

One of the first dedicated method and still widely used for establishing orthology relationships between two genomes is called Bi-directional Best Hit (BBH), which is based on the assumption that genes in different genomes that are the reciprocal best hit of each other are orthologs. A short-coming of this method that it is capable to infer one-to-one orthologous relationships. If duplications occur in any of the compared genomes, one-to-many or many-to-many relationships become necessary to fully describe the evolutionary relationships. In this case, Bi-directional Best Hit misses orthologs, more specifically co-orthologous predictions (see Figure 1.2). Despite its simplicity and *incomplete* inference, BBH outperforms some more complex orthology methods (Overbeek et al., 1999; Kristensen et al., 2011; Altenhoff and Dessimoz, 2009). Several methods use the Bi-directional Best Hit approach as part of

the graph construction phase and extend it to multiple genome comparisons in the clustering phase.

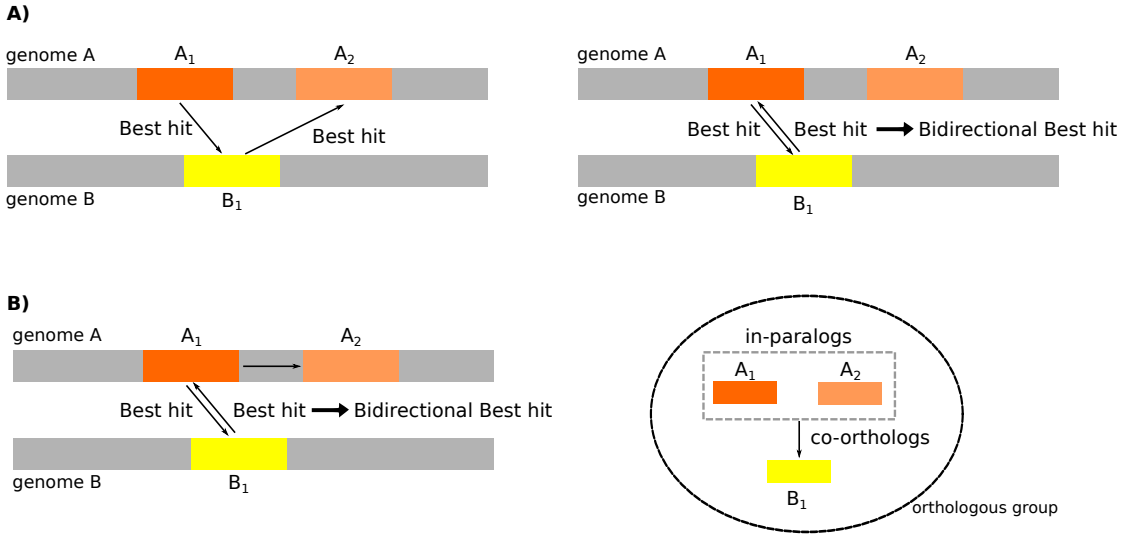



**Figure 1.2:** Graph-based methods. **A)** Bi-directional best hit (BBH). Only pairs of genes that are reciprocal best hits are considered orthologs. On the left no ortholog is assigned, on the right $A_1$ and $B_1$ are considered orthologs. Note that if $A_1$ and $A_2$ are in-paralogs, both would be co-orthologs of $B_1$, but since BBH is only capable to infer one-to-one relationships, it does not consider $A_2$ as an ortholog, giving rise to a miss prediction. **B)** Inparanoid approach. This is similar to A) but other genes within the genome ($A_2$ in this example) are included as in-paralogs if they are more similar to each other than to their corresponding hits in the other species. We end up with an orthologous group (on the right), where the genes can either be orthologs if the genes belong to different species, or in-paralogs if the genes belong to the same species.

Graph-based methods are not phylogeny-aware and therefore in principal incapable of distinguishing the different sub-types of paralogy. Moreover they have been shown to be less accurate on average (Kuzniar et al., 2008; Trachana et al., 2011a). Nonetheless, they are less costly in terms of computational resources and therefore the method of choice for large-scale analyses.

**Table 1.1:** Overview of graph-based orthology inference methods and their main properties

| Graph-based methods | | | | | |
|---|---|---|---|---|---|
| **Methods & Databases** | **In-paralogs** | **Homology search** | **Clustering strategy** | **Coverage** | **Reference** |
| **COG** | Yes | BLAST | Merged adjacent triangles of Best Hits | ALL | Tatusov, 1997 |
| **BBH** | No | BLAST | *n.a.* | ALL | Overbeek et al., 1999 |
| **InParanoid** | Yes | BLAST | Only between pairs of species | ALL | Remm et al., 2001; Ostlund et al., 2010 |
| **RSD** | No | ML distance estimates | *n.a.* | ALL | Wall et al., 2003 |
| **OMA** | Yes | ML distance estimates | Every pair is ortholog | ALL | Dessimoz et al., 2005; Altenhoff et al., 2011 |
| **OrthoMCL** | Yes | BLAST | MCL clusters | ALL | Li et al., 2003; Chen et al., 2006 |
| **eggNOG** | Yes | BLAST | Merged adjacent triangles of Best Hits | ALL | Jensen et al., 2008; Powell et al., 2012 |
| **OrthoDB** | Yes | Swiss-Waterman | Merged adjacent triangles of Best Hits | Eukaryotes | Kriventseva et al., 2008 |

### 1.4.2 Tree-based methods

Tree-based methods infer orthologous and paralogous relationships from phylogenetic trees. The general approach can be divided into two main steps: build the gene tree and reconcile it with the species tree. For the first part, a set of proteins of different species is collected according to a given threshold of similarity - normally based on a BLAST e-value. Next, a multiple sequence alignment is constructed to establish homologous sites which serve as characters to infer the genealogy. In the second part, the gene tree is reconciled with the species tree in order to label each internal node as speciation or duplication event. Two examples of tree reconciliation are illustrated in Figure 1.3.

As a result, if we want to infer the evolutionary relationships of any two genes in the gene tree, we simply find the last common ancestor of the two. If it is a speciation event the genes are orthologous, if it is a duplication event the genes are paralogous (Kristensen et al., 2011; Kuzniar et al., 2008).

By considering all sequences jointly, tree-based methods can extract more types of evolutionary events from the sequences such as gene loss and duplication events. Their phylogenetic framework allows the classification of genes into orthologs, in-paralogs, co-orthologs and out-paralogs; hence, they are more powerful. However, there are several challenges that this type of methods may have to overcome: multiple sequence alignment quality, rooting, gene tree uncertainty, unresolved species trees (Stevcic, 1978; van der Heijden et al., 2007). Moreover, these approaches are computationally intensive and are normally performed only on small sets of species.

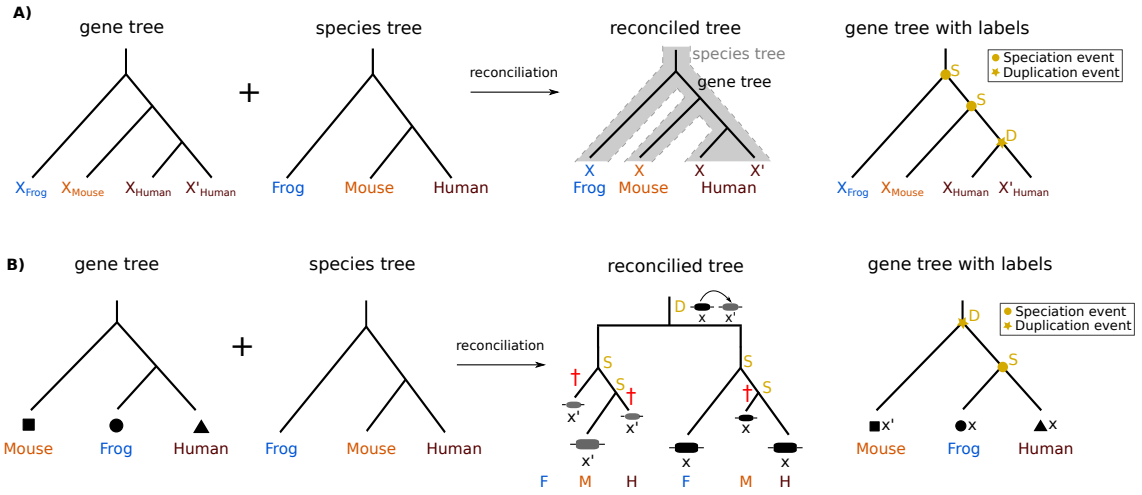



**Figure 1.3:** Tree reconciliation phylogenetic approach. Duplication nodes ($D$) are defined by comparing the gene tree with the species tree, resulting in a reconciled tree in which the minimal number of duplication and gene loss events necessary to explain the gene tree are included. **A)** Simple illustration of tree reconciliation. Given that gene $x'$ is only present in Human, and not in Mouse for example, the most parsimonious explanation is that a duplication occurred after the second speciation event. **B)** A more elaborate example of tree reconciliation. Since the gene (●) in frog clusters with the human gene (▲) suggesting they are more similiar, the most parsimonious assumption is that there was a duplication event ($x$ to $x'$) before all the speciation events occurred, followed by two gene losses ($x'$) in Frog and Human and a gene loss ($x$) in Mouse.

**Table 1.2:** Overview of tree-based orthology inference methods and their main properties

| Tree-based methods | | | | |
|---|---|---|---|---|
| **Methods** | **Homology search** | **Gene Tree** | **Coverage** | **Reference** |
| Ensembl Compara | all vs. all WUBlastp + Smith-Waterman + clustering | TreeBeST | Vertebrates | Vilella et al., 2009 |
| PhylomeDB | seed vs. all Smith-Waterman | BioNJ, PhyML, MrBayes | ALL | Huerta-Cepas et al., 2008, 2011a |
| TreeFam | Blast & HMMER | TreeBeST | Metazoa | Li et al., 2006; Ruan et al., 2008 |

In summary, the comparison between graph-based and tree-based methods reveals a clear trade-off between accuracy on one hand and efficiency and coverage on the other. Hence, no method is clearly better than other and it rather depends on the purpose of the orthology search. For example, if the aim is to find an ortholog in a certain species, the best is to choose a more specific method, on the other hand if the aim is to find orthologs across several species the best choice is a more sensitive method. Other factors are the availability of the orthologs, different methods have different species coverage, *i.e.* if the aim is to find orthologs in plants, a method that only covers animal species can not be used; how well the relationships between species are known, for example for methods that use the species tree (majority of tree-based methods) the uncertainty between some species can become a limiting factor for an accurate inference (Gabaldón, 2008).

Our ambition was to get the best of both worlds and we therefore aimed to develop an algorithm that combines the efficiency of large-scale pairwise methods and the accuracy of small-scale phylogenetic approaches.

## 1.5 Overview

In the next chapter we present TreeHop, the algorithm that was developed for this thesis and that aims to improve current orthology inference methods. In chapter 3 we show how we validated the algorithm and how it performs in general. In chapter 4 we analyse possible variables that can affect TreeHop performance and in chapter 5 we try to optimize the algorithm using different heuristics. The conclusion and future perspectives are presented in the last chapter.

We show that TreeHop is able to find more orthologs without reducing the overall quality of predictions. We identify and validated different heuristics able to modulate the performance between many/few predictions or high/low quality predictions.

# 2 TreeHop algorithm and its implementation

As seen in the previous chapter, there is still need for an accurate yet efficient orthology inference method. The fundamental assumption on which TreeHop relies is that orthology inference is easier between closely related species than between distantly related species. This is supported by basic evolutionary considerations: closely related species are expected to share more orthologs and have less divergent sequences.

We first confirmed this assumption by collecting a set of species at different taxonomic distances and inferring their orthologs by Bi-directional Best Hit. As shown in Figure 2.1, we indeed observe more orthologs between closely related species, specially among vertebrates and fungi. The same overall pattern can be observed for sequence similarity (see Figure 2.2).

Note that two leaves that are close in the tree taxonomy might be separated by a low taxonomic rank or by a high taxonomic rank, which is represented by the different colors of the branches.

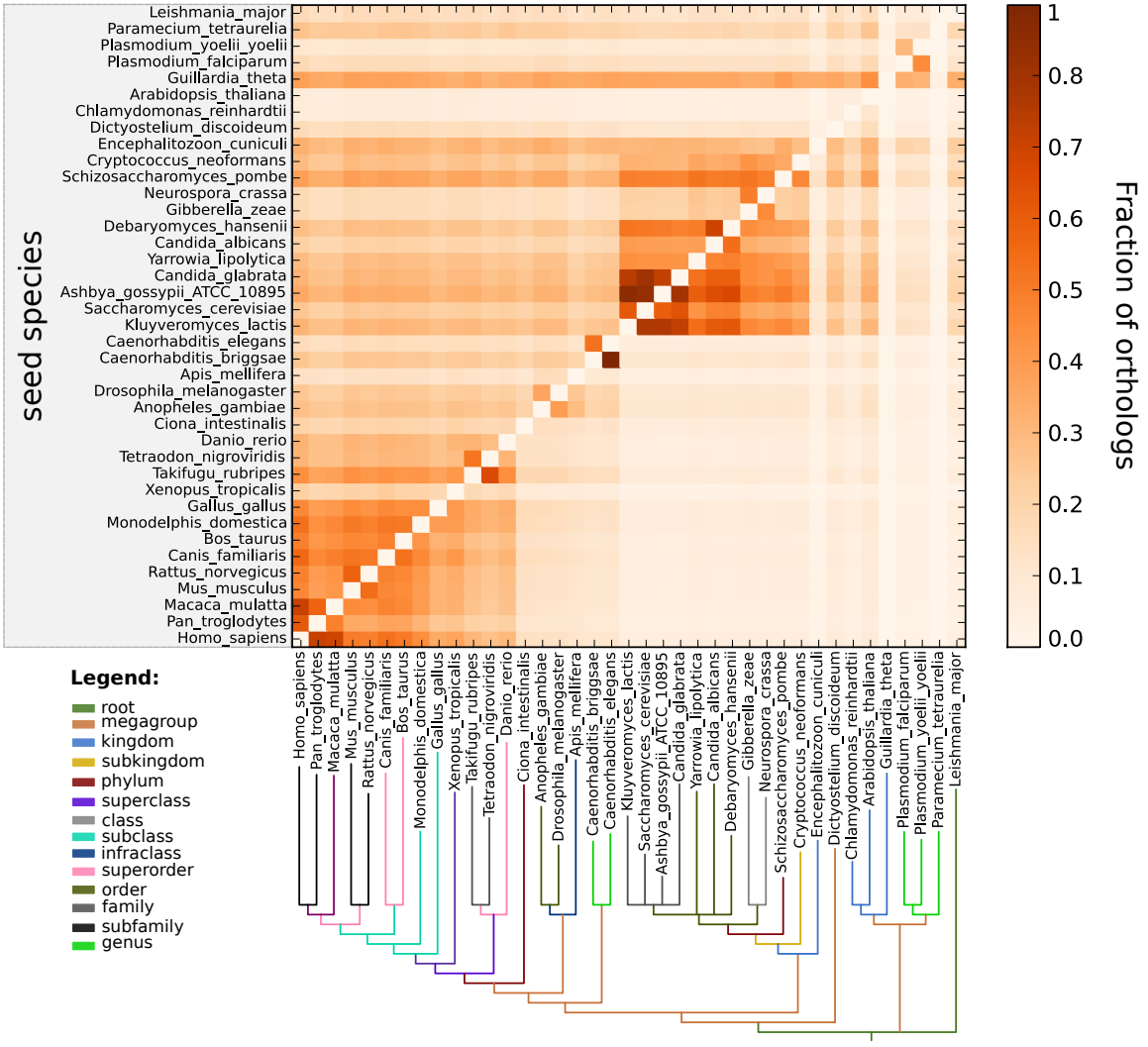



**Figure 2.1:** Fraction of shared orthologs between species. The different branches are coloured according to the corresponding taxonomic rank.

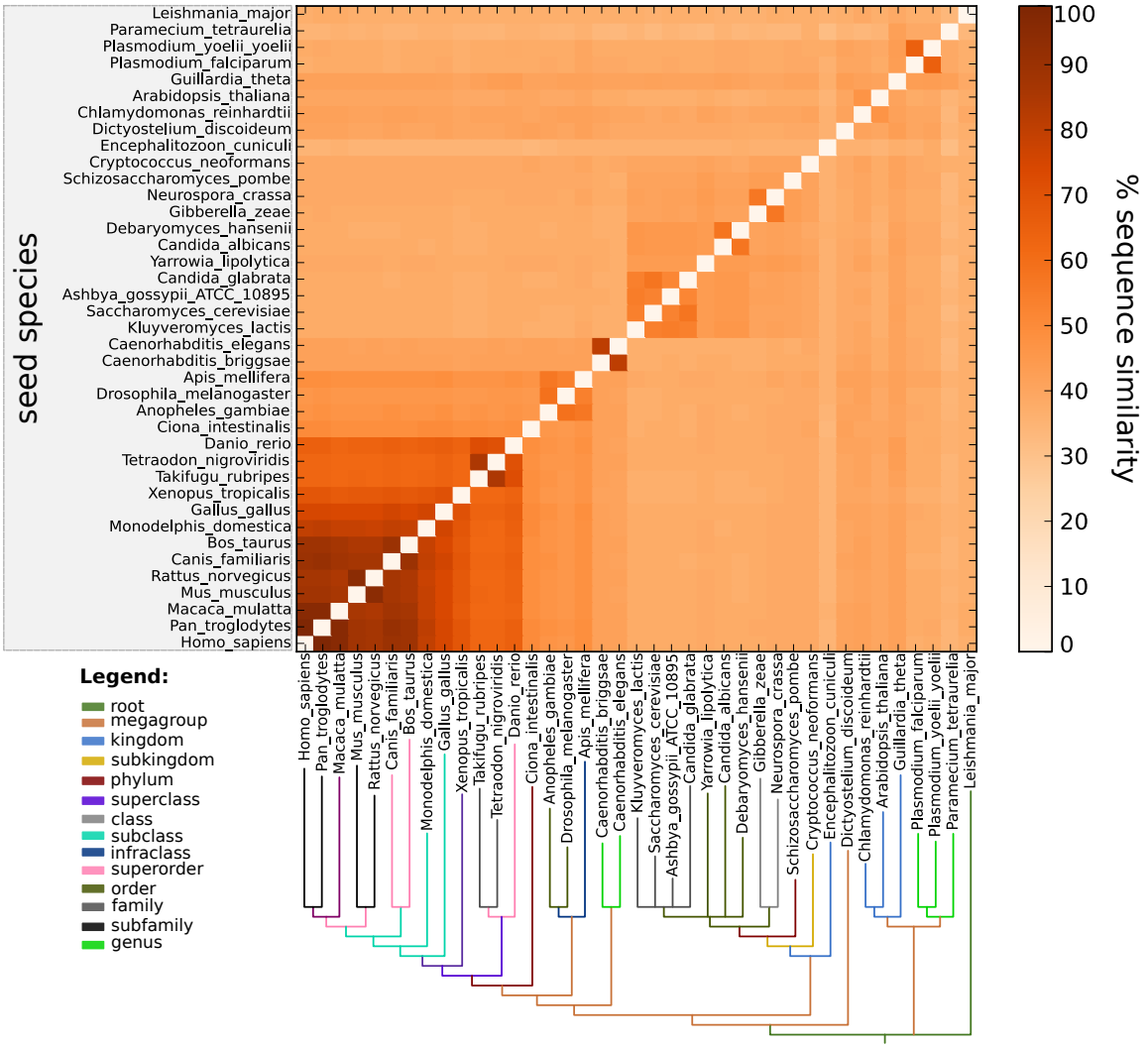



**Figure 2.2:** Percentage of sequence similarity between species. The gradient corresponds to high sequence similarity (dark orange) and high sequence divergence (white). The different branches are coloured according to the corresponding taxonomic rank.

## 2.1 Basic approach

In order to improve the orthology detection across distantly related species we designed an algorithm that resembles hopping between branches: TreeHop.

TreeHop requires a set of proteomes, the corresponding species tree and a profile of orthologs detected by an existing orthology inference method. The aim is to find orthologs which may have been missed in the given profile. To this end, TreeHop exploits the orthologs found in the closest species as a starting point for new orthology predictions. A graphical representation of the basic approach is shown in Figure 2.3.

We defined the following terms which are used throughout the remainder of the thesis:

- **base-method:** orthology detection method used as a starting point for TreeHop performance
- **gap species or gap:** species in which the base method did not detect an ortholog
- **seed gene:** gene used as input to the base method to detect orthologs
- **seed species:** species to which the seed gene belongs
- **source gene:** gene used by TreeHop as an input to jump
- **source species:** species to which the source gene belongs

For example, in Figure 2.3 a given orthology detection method assigned orthologous genes in species B and C for gene $x$ in species A. No orthologs were found by that method in species D and E. For each species without an ortholog, for example species D, TreeHop traverses the tree and finds the closest species for which an ortholog has been assigned, in this case, species C. The predicted ortholog in species C is used as source gene to predict an ortholog in species D. If after the first jump (from species C to D) no ortholog is detected, TreeHop continues traversing the tree visiting the nodes by order of distance from the gap species and stops hopping when it finds an ortholog or when there are no more species to jump from. Notice that TreeHop's output is composed of orthologs found by the base method and orthologs found by TreeHop.

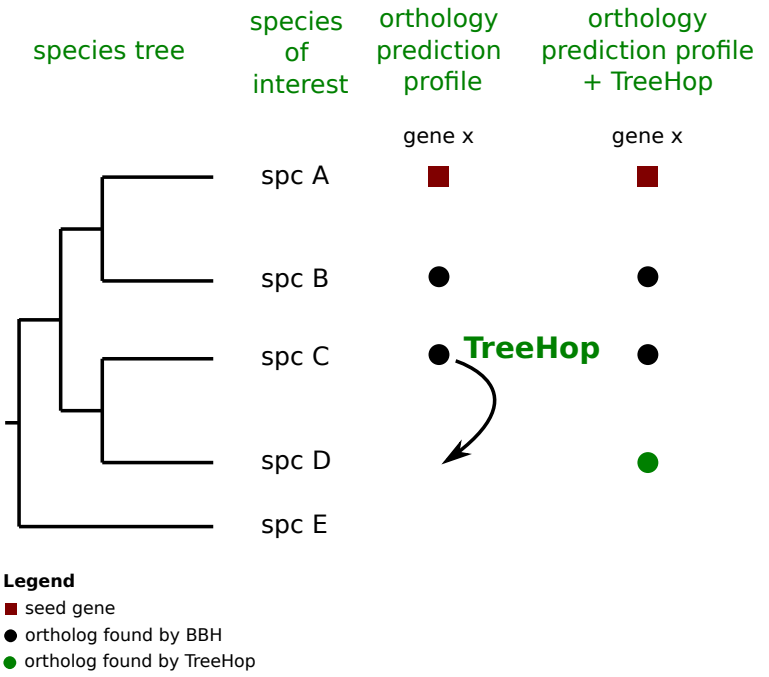



**Figure 2.3:** TreeHop basic approach and possible output. TreeHop takes as input a set of proteomes, the corresponding species tree and a profile of orthologs (black circles) detected by an existing orthology inference method (base method). For each gap species (species with no ortholog detected by the base method), TreeHop uses the ortholog detected in the closest species to find a new ortholog. In this example, TreeHop uses the gene in species C detected by the base method as a starting point to find an ortholog (green circle) in species D. This jump from the gene (source gene) in species C to species D is indicated with a black arrow. Another possible output would be the detection of another ortholog in species E or no detection in both or one of the gap species.

## 2.2 Implementation

TreeHop can use any existing orthology detection method (graph or tree-based) as base method and is prepared to use any pairwise method or a simple BLAST to jump between species. Note that the base method and the method that is used to jump are independent, *i.e.* they can be the same or different. Here we chose to use the method Bi-directional Best Hit (BBH) as base and jump method to find additional orthologs.

The flowchart in Figure 2.4 summarizes TreeHop. The input is a species tree and a profile of orthologs detected by the base method for a certain seed gene. For each gap (species B and D in the example), TreeHop jumps from the closest species to the gap in order to find an ortholog. If it does not find an ortholog in the first jump it continues traversing the tree in that order. It only stops jumping if it finds an ortholog in the correspondent gap or if there are no species left to jump from.

For a given set of $n$ species, generating an input profile of orthologs requires to compute $2(n-1)$ Best Hits (2 per BBH). For a profile with $m$ orthologs (to the seed gene), with $0 < m < n-1$, TreeHop is going to perform at most $2(n-1-m)m$ additional BH computations, leading to an overall quadratic worst case behaviour. However, note that in practice this

is rarely going to be the case. In comparison, all vs. all BBH always requires a number of comparisons quadratic in $n$, and an additional clustering step. Furthermore, all vs. all BBH cannot be performed only for a subset of proteins in a genome, rather always requires to do the analysis genome-wide. Hence, compared to graph-based methods, TreeHop is not only expected to be more efficient in practice, but also allows to restrict the computations only to the proteins of interest.

We used a taxonomic species tree, retrieved from the NCBI taxonomy[1] (see Figure 7.2). In general, taxonomic trees suffer from two drawbacks: first, no branch lengths are defined, and second, they may contain polytomies. For our purposes these turned out not to be problematic, as we defined distance solely based on tree topology. Also, the resulting tree for our set of species is well resolved and only contains 3 polytomies.

[1] `http://www.ncbi.nlm.nih.gov/taxonomy`

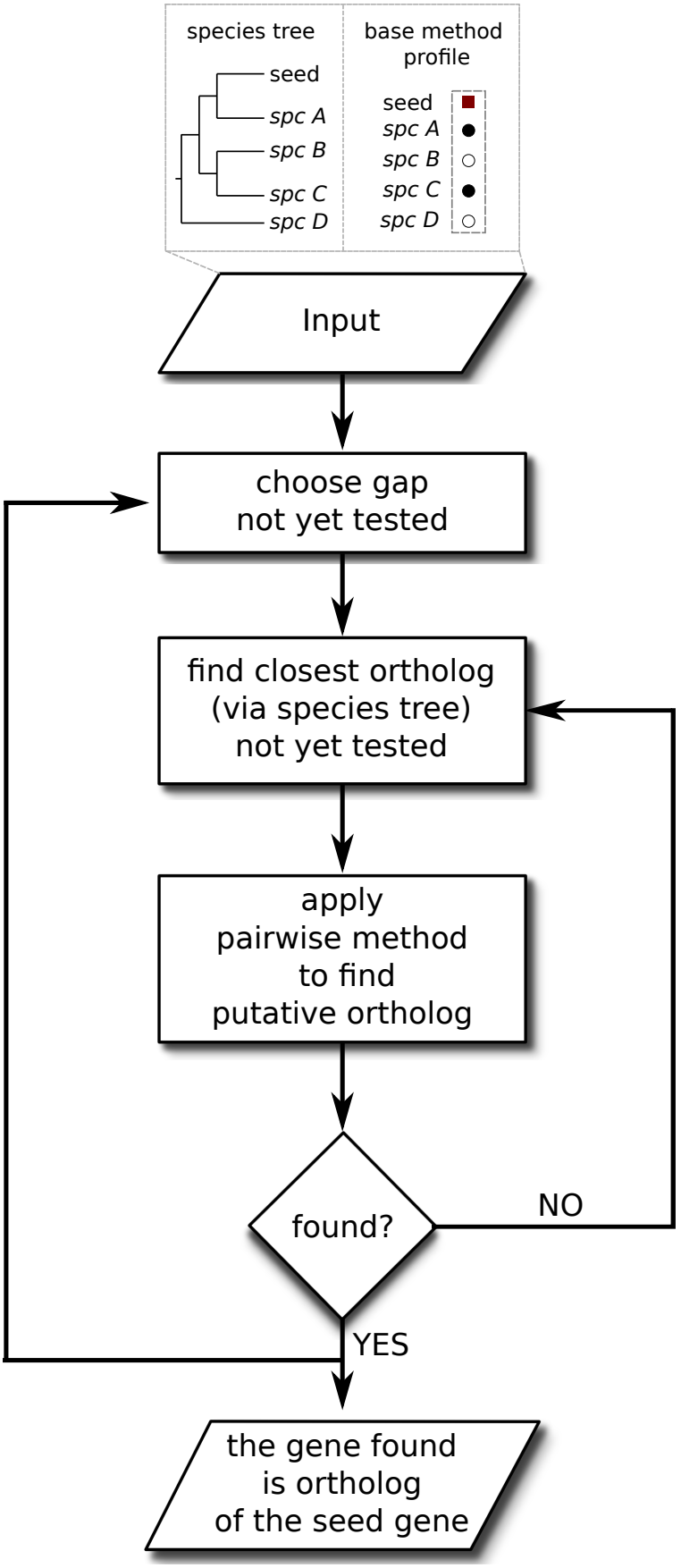



**Figure 2.4:** TreeHop flowchart.

# 3 TreeHop performance and validation

## 3.1 Validation of orthology detection

Every predictive method must be validated against a gold standard. For orthology prediction in particular, this is complicated by a lack of a gold standard that satisfies the requirement of quality and large scale. Moreover, the evolutionary history of genomes is unknown and since we do not have access to the ancestral genomes, we can never be sure that the assumption of two proteins being orthologous/paralogous/homologous is indeed correct.

### How do we deal with the absence of a large-scale gold standard?

The validation of an orthology inference method can be done by comparing it to a gold set which results either from a phylogeny aware method, or from a manually curated set of protein families or even simulated protein families.

The first two ways, although accurate, lack coverage *i.e.* the number of species that they cover is small. There has been an effort to extend the number of species in phylogenetic-based methods, but for each new species added, all gene trees have to be recalculated, which is computationally intense.

Simulations are virtually the only way to directly obtain information about ancestral sequences. Different tools exist that mimic the evolution of an initial sequence by simulating sequence divergence along a given tree. They provide the option to include evolutionary phenomena, such as: duplication, gene loss, change in GC content, point mutations, indels, etc. Although this strategy can give us a set of true orthologs to compare with, it lacks the ability to reflect the full complexity of gene family evolution observed in nature. Thus, this strategy is can be used as a complementary approach to validate orthology inference methods (Storm and Sonnhammer, 2002; Altenhoff et al., 2013; Dalquen et al., 2013).

To overcome the absence of a gold standard and in order to evaluate the orthology detections among different methodologies, some authors proposed to do a Latent Class Analysis. This is a a purely statistical benchmark which looks at agreements (enhances confidence) and disagreements (indicates possible errors) of predictions made by several methods on a common dataset (Chen et al., 2007).

Here, we present the validation of TreeHop against a large-scale dataset, using the phylogenetic-aware method PhylomeDB (Huerta-Cepas et al., 2007) and a small-scale dataset, using a set of 70 manually curated families (Trachana et al., 2011b).

## 3.2 TreeHop validation

The predictions made by a certain method, when compared to a gold standard, are classified into values from the confusion matrix: True Positives (TP) if the method output is in agreement with the gold standard or False Positives (FP) if the method output is in disagreement with the gold standard. If the method did not make predictions, this can be classified into: True Negatives (TN) if the gold standard has also no ortholog assigned or False Negatives (FN) if the gold standard has at least one ortholog assigned.

The following properties are going to be important to understand the validation of the algorithm: TreeHop depends on the orthologs detected by the base method and all the orthologs detected by the base method are part of TreeHop's output. After TreeHop's performance the number of orthologs detected are either: the same as it started with, meaning that TreeHop did not find any other ortholog; or increase, meaning that TreeHop found at least one more ortholog. This means that if the base method detects $x$ orthologs, after TreeHop's performance the number of orthologs detected in the ouput can only be $x$ or bigger than $x$. Note that TreeHop cannot decrease the number of orthology detections made by the base method.

Again, since TreeHop acts on top of an existing orthology detection method (base method) inheriting its orthologs, the fact that the base method has true predictions, false predictions and/or miss predictions, is going to influence the final TreeHop outcome. Given this nested structure, it is important to understand how the values from the confusion matrix TP, TN, FP and FN are obtained. Figure 3.1 illustrates how this works: the predictions made by the base method can either be true predictions (TP) or false predictions (FP). Note that this is always going to be part of the final TreeHop's output, since the algorithm cannot "remove" the predictions already made by the base method. When the base method does not predict orthologs this can either be a true negative(TN) or a miss-prediction (FN). Since the lack of prediction corresponds to the *gaps* that TreeHop tries to fill in, these are the values that might change. If TreeHop finds an ortholog where a TN has been previously assigned by the base method, this prediction can only be false (FP). On the other hand, if TreeHop finds an ortholog where a FN has been assigned, this can either become a TP if the ortholog found is also present in the gold standard or a FP if the ortholog found is not present in the gold standard.

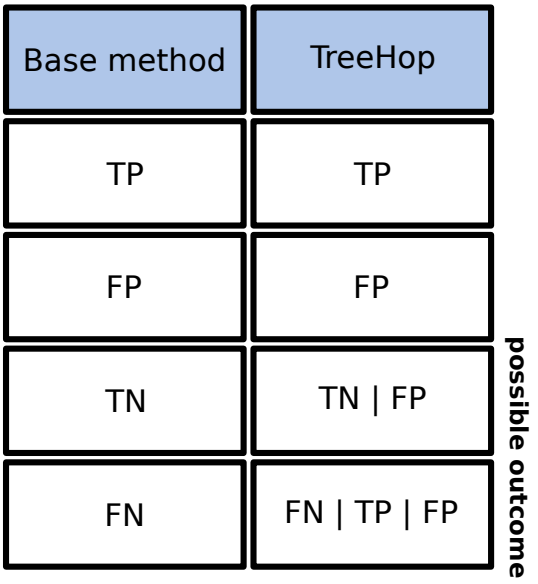


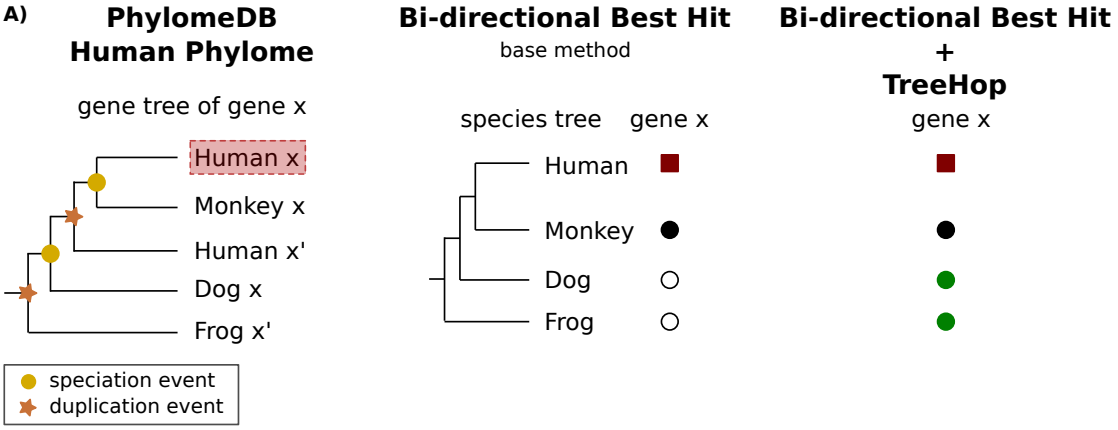

**Figure 3.1:** Confusion matrix values of the base method and TreeHop. The TP and FP obtained by the base method are automatically part of TreeHop's output, the TN and FN can either remain the same if TreeHop does not predict anything, or change to a FP and TP or FP, respectively, if TreeHop predicts an ortholog.

### 3.2.1 Validation against PhylomeDB

Since the aim of our algorithm is to detect orthologs on a large-scale, we chose PhylomeDB, more specifically the Human Phylome. First because of its phylogenetic framework that provides accurate results, and second due to its comparatively large coverage across different eukaryotic branches.

A phylome is a collection of evolutionary histories of all genes in a genome (Huerta-Cepas et al., 2008). In the case of the Human Phylome, for each gene in the human genome, the authors computed gene trees from homologous sequences from 38 Eukaryotes. Each human gene was used as the seed to find homologous sequences in other species, to compute its correspondent gene tree and to infer a set of paralogous and orthologous sequences.

Considering that we are only interested in improving orthology detection, we focused on the orthologs, which correspond to pairs of leaves in the gene tree whose last common ancestor is an internal node annotated as speciation event. This is illustrated in Figure 3.2.




B)

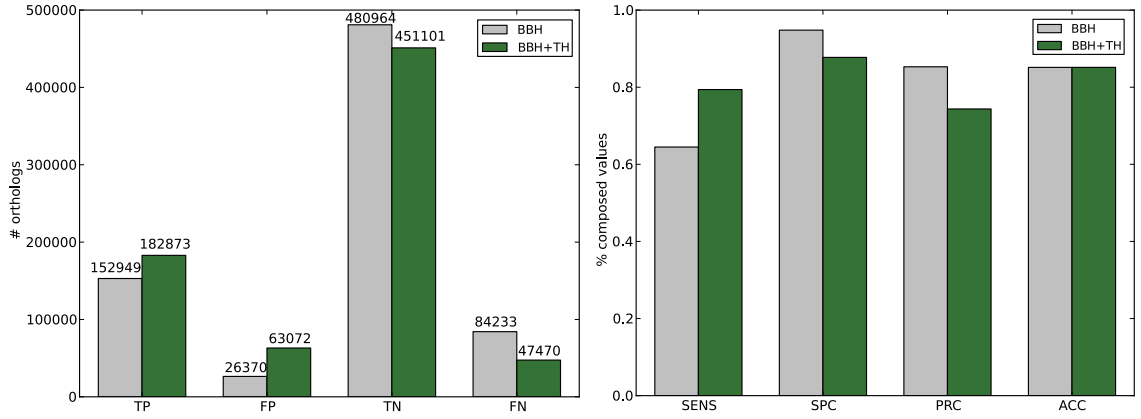
| | BBH | TreeHop | BBH + TreeHop |
|---|---|---|---|
| TP | 1 | 1 | 2 |
| FP | 0 | 1 | 1 |
| TN | 1 | -1 | 0 |
| FN | 1 | -1 | 0 |

**Figure 3.2:** Example of BBH (base method) and TreeHop validation against PhylomeDB. **A)** Example of gene x from human used as seed (pink square) to find homologous sequences in a set of species (Human, Monkey, Dog and Frog). The same seed was used to detect orthologs using the base method BBH followed by TreeHop. BBH does not find any ortholog in Dog and Frog, but TreeHop does. **B)** Table of the confusion matrix values – True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN). Considering that we are only interested in finding orthologous genes, only the genes for which the internal node with the seed are a speciation event are considered, *i.e.*, gene $x$ in monkey and dog only. Gene $x'$ in frog and human are considered paralogous to the seed so are not considered. BBH finds the gene $x$ in Monkey - TP, does not make wrong predictions - no FP, misses the prediction of gene $x$ in Dog - FN and does not find any ortholog in frog, which is correct - TN. TreeHop on top of that finds gene $x$ in Dog - TP but also finds gene $x'$ which is a paralog, not an ortholog - FP.

### TreeHop's performance

The basic TreeHop performance when validated against PhylomeDB is illustrated in Figure 3.3. The graph shows the performance of the base method Bi-directional Best Hit (BBH) alone and BBH + TreeHop. TreeHop increases the number of true predictions (TP) but also the number of false predictions (FP), when compared with the base method. It also decreases the number of miss-predictions (FN) and the number of true negatives (TN). In terms of derived measures of the confusion matrix, TreeHop improves the sensitivity (SENS) and accuracy (ACC) of the base method and decreases its specificity (SPC) and precision (PRC).

A)




B)

| | BBH | BBH + TreeHop | TreeHop |
|---|---|---|---|
| TP | 152949 | 182873 | 29924 |
| FP | 26370 | 63072 | 36702 |
| TN | 480964 | 451101 | -29863 |
| FN | 84233 | 47470 | -36763 |
| SENS | 0.645 | 0.794 | 0.149 |
| SPC | 0.948 | 0.877 | -0.071 |
| PRC | 0.853 | 0.744 | -0.109 |
| ACC | 0.851 | 0.852 | 0.00008 |

**Figure 3.3:** Bidirectional Best Hit (BBH) and TreeHop (TH) performances, using PhylomeDB as the gold standard. TP - true positives (hit), FP - false positives (false alarm), TN - True Negatives (correct rejection), FN - false negatives (miss). SENS - sensitivity (hit rate or recall), SPC - specificity (true negative rate), PRC - precision (positive predictive value), ACC - accuracy. Sensitivity = $\frac{TP}{TP+FN}$, specificity = $\frac{TN}{TN+FP}$, precision = $\frac{TP}{TP+FP}$ , accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$. **B)** Table with the raw values from the confusion matrix accoding to the performances of BBH, BBH + TreeHop and TreeHop alone. The values of the third column are illustrated in Figure 3.4.

As discussed earlier in section 3.2 the overall performance includes the base method performance. As we wish to focus solely on TreeHop, we illustrate its performance alone in Figure 3.4. Here, the graph shows absolute values of what treeHop increases and decreases in a standardized manner. However, here an increase not always means improvement. The pink bars represents what TreeHop improves, *i.e.* the gain in TP and the decrease in FN, the increase in sensitivity ($\approx 15\%$) and accuracy ($\approx 0.08\%$); whereas the grey bars represent what TreeHop diminishes, *i.e.* the gain in FP, the decrease in TN, precision ($\approx 11\%$) and specificity ($\approx 7\%$).

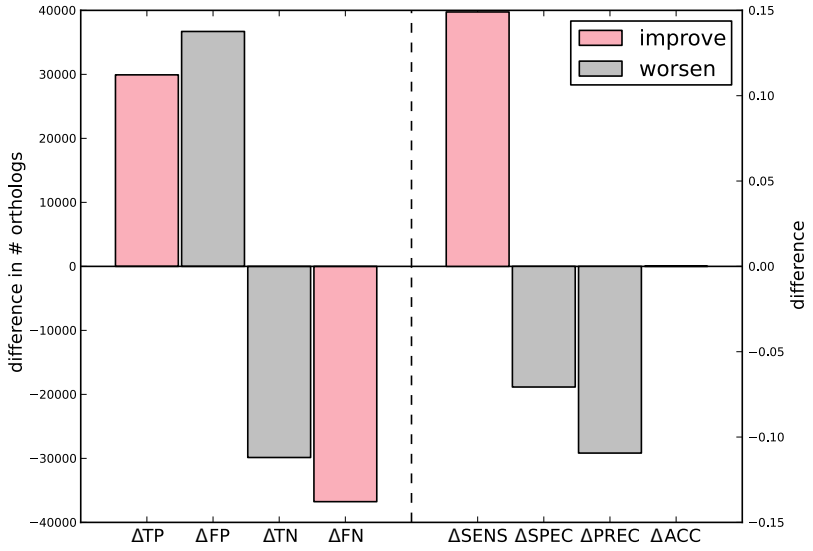



**Figure 3.4:** TreeHop performance. Graph showing what values TreeHop increases (positive y axis), or decreases (negative y axis). It also shows what TreeHop improves (pink bars) and what it diminish (grey bars). The Δ represents the subtraction between the base method + Treehop by the base method values. $\Delta TP = TP_{\text{base method+TreeHop}} - TP_{\text{base method}}$, $\Delta FP = FP_{\text{base method+TreeHop}} - FP_{\text{base method}}$, $\Delta TN = TN_{\text{base method+TreeHop}} - TN_{\text{base method}}$, $\Delta FN = FN_{\text{base method+TreeHop}} - FN_{\text{base method}}$, $\Delta SENS = \text{Sensitivity}_{\text{base method+TreeHop}} - \text{Sensitivity}_{\text{base method}}$, $\Delta SPC = \text{Specificity}_{\text{base method+TreeHop}} - \text{Specificity}_{\text{base method}}$, $\Delta PRC = \text{Precision}_{\text{base method+TreeHop}} - \text{Precision}_{\text{base method}}$, $\Delta ACC = \text{Accuracy}_{\text{base method+TreeHop}} - \text{Accuracy}_{\text{base method}}$.

### Conclusion

TreeHop predicts more true positives. However, the classic trade-off is observed *i.e.* the gain of true positives at the expense of an increase in false positives. Yet, the accuracy is higher, meaning that TreeHop increases the base method's overall performance. This can be useful in the case where the standard analysis does not make predictions for the species of interest. We can apply TreeHop (with the necessary caution) to at least end up with orthologs in that species. If TreeHop is also not able to find an ortholog, we can be more confident that actually there is none.

### 3.2.2 Validation against 70 manually curated families

We validated TreeHop against a publicly available[1] set of 70 manually curated proteins families. This is a small-scale dataset which only comprises 12 species, yet it is relevant due to its manual curation and, more importantly, its selected families for exploring caveats of orthology prediction: rate of evolution, lineage-specific loss/duplication and alignment quality. This provides us with a platform to test TreeHop overall performance using the full dataset, and TreeHop's specific performance according to different orthology challenges.

Figure 3.5 illustrates TreeHop's performance on the full dataset. We observe that it improves the base method sensitivity by ≈ 17% and its accuracy by ≈ 5%; it also decreases ≈ 6% of precision and ≈ 20% of specificity.

**A)**

| | BBH | BBH + TreeHop | TreeHop |
|---|---|---|---|
| TP | 751 | 918 | 167 |
| FP | 191 | 333 | 142 |
| TN | 335 | 266 | -69 |
| FN | 483 | 265 | -218 |
| SENS | 0.609 | 0.776 | 0.167 |
| SPC | 0.637 | 0.444 | -0.193 |
| PRC | 0.797 | 0.734 | -0.063 |
| ACC | 0.617 | 0.664 | 0.047 |

**B)**

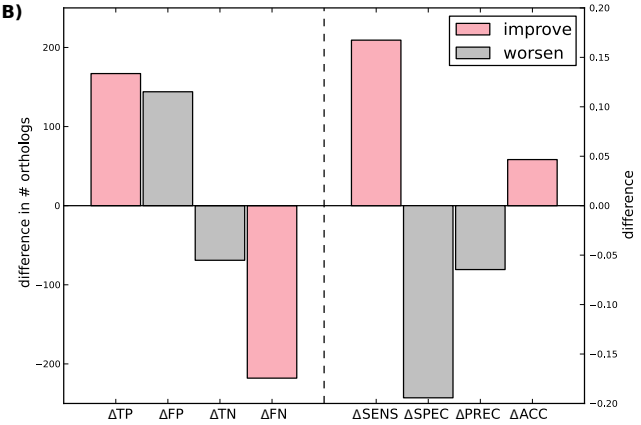



**Figure 3.5:** TreeHop performance against the 70 manually curated families. **A)** Table with the raw values from the confusion matrix accoding to the performances of BBH, BBH + TreeHop and TreeHop alone. The values of the third column are illustrated in panel B. **B)** Graph showing what values TreeHop increases (positive y axis) or decreases (negative y axis). It also shows what TreeHop improves (pink bars) and what it diminish (grey bars). The Δ represents the subtraction between the base method + Treehop by the base method values. $\Delta TP = TP_{\text{base method+TreeHop}} - TP_{\text{base method}}$, $\Delta FP = FP_{\text{base method+TreeHop}} - FP_{\text{base method}}$, $\Delta TN = TN_{\text{base method+TreeHop}} - TN_{\text{base method}}$, $\Delta FN = FN_{\text{base method+TreeHop}} - FN_{\text{base method}}$, $\Delta SENS = \text{Sensitivity}_{\text{base method+TreeHop}} - \text{Sensitivity}_{\text{base method}}$, $\Delta SPC = \text{Specificity}_{\text{base method+TreeHop}} - \text{Specificity}_{\text{base method}}$, $\Delta PRC = \text{Precision}_{\text{base method+TreeHop}} - \text{Precision}_{\text{base method}}$, $\Delta ACC = \text{Accuracy}_{\text{base method+TreeHop}} - \text{Accuracy}_{\text{base method}}$.

The following figures illustrate TreeHop's performance according to different orthology challenges: rate of evolution (fast- vs. slow-evolving families) in Figure 3.6, lineage-specific loss/duplication (single copy families vs. multiple duplication events) in Figure 3.7, and alignment quality (high- vs. low-quality alignment) in Figure 3.8.

Regarding the three categories previously analysed, the results are in agreement with the authors' observations (Trachana et al., 2011b). TreeHop performs better for slow evolutionary rates, it increases 13% of the accuracy and 20% of sensitivity of the base method. Fast-evolving families tend to accumulate a larger number of errors, and we expect worse performances. TreeHop also presents a better performance when dealing with small families,

[1] `http://eggnog.embl.de/orthobench`

meaning less paralogs, which makes sense given that the method used to jump (BBH) can only get one-to-one orthology relationships, and is not able to detect in-paralogs. Regarding the alignment quality, TreeHop performs better on high quality alignments, increasing 7% of the accuracy and 15% of the sensitivity of the base method.

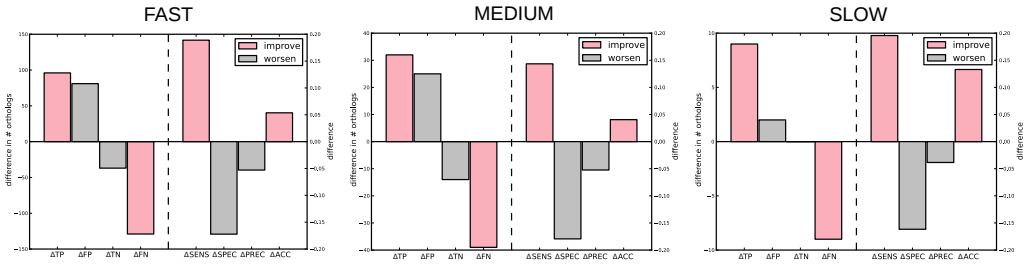



**Figure 3.6:** TreeHop performance at different rates of evolution. Graph showing what values TreeHop increases (positive y axis) or decreases (negative y axis). It also shows what TreeHop improves (pink bars) and what it diminish (grey bars). The Δ represents the subtraction between the base method + Treehop by the base method values.

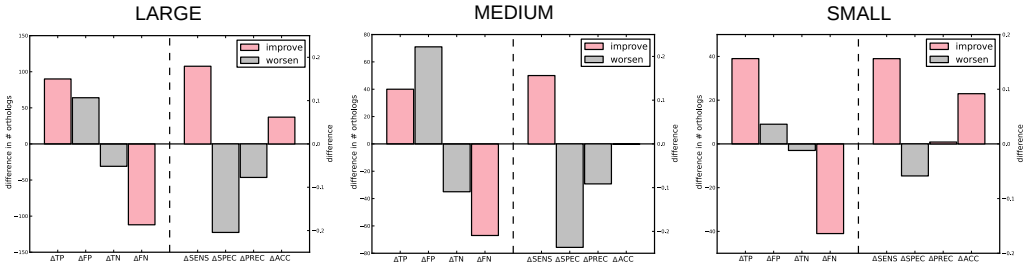



**Figure 3.7:** TreeHop performance at different family sizes (impact of duplication events - paralogs). Graph showing what values TreeHop increases (positive y axis) or decreases (negative y axis). It also shows what TreeHop improves (pink bars) and what it diminish (grey bars). The Δ represents the subtraction between the base method + Treehop by the base method values.

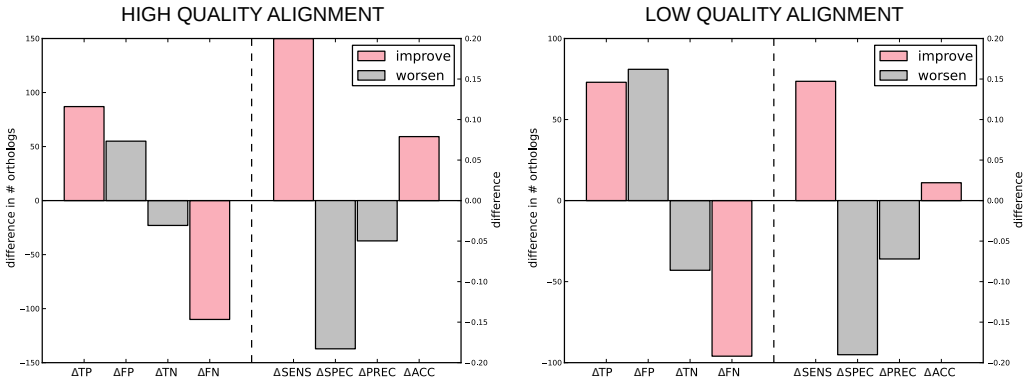



**Figure 3.8:** TreeHop performance at high and low multisequence alignment. Graph showing what values TreeHop increases (positive y axis) or decreases (negative y axis). It also shows what TreeHop improves (pink bars) and what it diminish (grey bars). The Δ represents the subtraction between the base method + Treehop by the base method values.

**Conclusion**

Comparable to the previous validation on a large-scale, TreeHop predicts more true positives. We again observe the trade-off between the increase of the number of TP at the expense of the number of FP, although the number of TP in this case exceeds the number of FP. Moreover, TreeHop highly increases the accuracy of the base method.

From the analysis against the different sub-sets of proteins, we can conclude that TreeHop's performance substantially increases with certain proteins *properties*, which suggests that it should be specially applied to, for example, small protein families with slow evolutionary rates.

TreeHop presents a much better overall performance against the small-scale dataset of manually curated families than against the large-scale dataset PhylomeDB. However, since our aim is to detect orthologs on a large-scale, we are going to focus only on TreeHop's performance against the gold standard PhylomeDB. Consequently, the next chapters are going to be dedicated to understand the possible factors that can affect the algorithm performance and to identify several parameters to optimize it.

# 4 Robustness of TreeHop performance

We wanted to understand the effect of different components on TreeHop performance. More specifically, the effect of the base method on TreeHop's performance, TreeHop's behaviour when using a wrong species tree and if there is any particular type of proteins for which TreeHop performs worse.

## 4.1 Effect of base method quality

TreeHop jumps from orthologs detected by the base method. To understand how this can influence the performance we inspected the source of the jumps from which TreeHop finds an ortholog. We wanted to know the fraction of true positives (TP) and false positives (FP) that came from a TP jump or a FP jump. The result is illustrated in Figure 4.1, and we observe that the majority of TreeHop's true predictions (bar on the left) come from TP jumps, whereas the majority of the false predictions (bar on the right) come from FP jumps. This lead us to hypothesize that the more specific the base method is, the better TreeHop is going to perform.

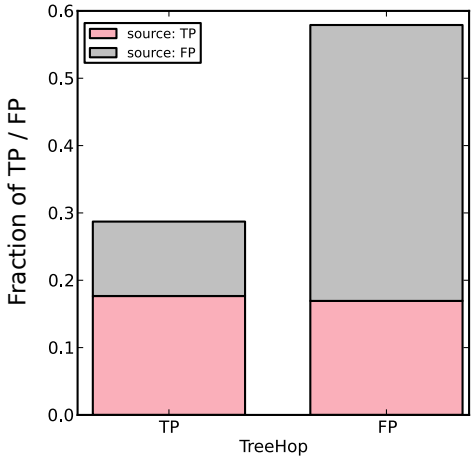



**Figure 4.1:** TreeHop predictions and their source. Each bar represents the True Positives and False Positives detected by TreeHop. In pink is the fraction of orthologs that came from a TP source gene and in grey is the fraction of orthologs that came from a FP source gene.

Testing this hypothesis requires the perfect set of "true" orthologs to jump from. Since this set does not exist on a large-scale, we decided to use our gold standard as the base method, which allows TreeHop to only jump from TP. We removed $\lfloor \frac{n}{2} \rfloor$ of the $n$ orthologs for each seed in the Human Phylome. To assess the effect of the base method alone we kept the Bi-directional Best Hit (BBH) as the jump method.

Note that the removal of 50% of the orthologs of the Human Phylome can only give us False Negatives, when compared this with the full set. The other remaining 50% necessarily correspond to TP. This experiment quantifies how many TP are recovered from the FN. As

shown in Figure 4.2, TreeHop recovers ≈ 95%, out of which ≈ 92% become TP and ≈ 8% FP. As a result, the sensitivity of the base method increases from 60% to 98% and the accuracy from 86% to 88%. The increased number of FP come from predictions in TN gaps, which decreases precision and specificity by 25% and 17%, respectively.

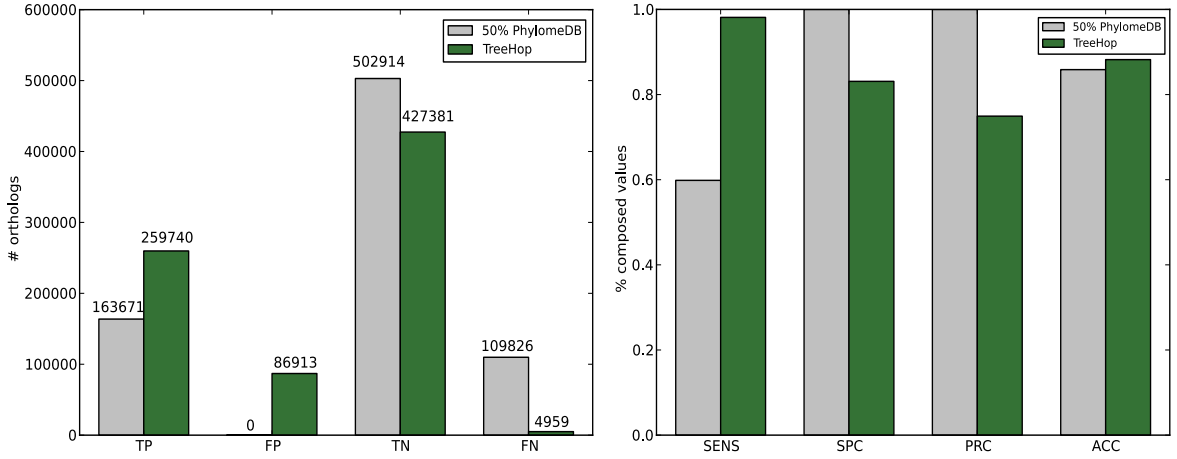



**Figure 4.2:** Human Phylome with 50% of the orthologs assignments removed and Bi-directional Best Hit + TreeHop performance.

This experiment confirms the hypothesis that a more specific base method improves TreeHop's overall performance.

## 4.2 Different species tree

TreeHop is guided by a species tree in order to find new putative orthologs. It jumps from the closest species to the gap and continues traversing the tree until it finds an ortholog. We asked how robust TreeHop is against misspecification of the species tree. We developed an experiment in which for each taxonomic rank (genus, family, class, phylum, etc...) the leaves are shuffled in order to measure the effect of the species tree on TreeHop's overall performance. For each taxonomic rank we repeated shuffling five times. Notice that due to hierarchical organization of the taxonomic ranks, shuffling a certain taxonomic rank also shuffles all its lower ranks. For example, shuffling the leaves at the taxonomic rank "class" the lower taxonomic ranks are also shuffled, in this case, "order", "family" and "genus".

In Figure 4.3 we observe that the number of TP detected by TreeHop decreases the higher the taxonomic rank is and the number of FP increases. Also, the loss or gain in TP and FP, respectively, is very small for lower taxonomic ranks.

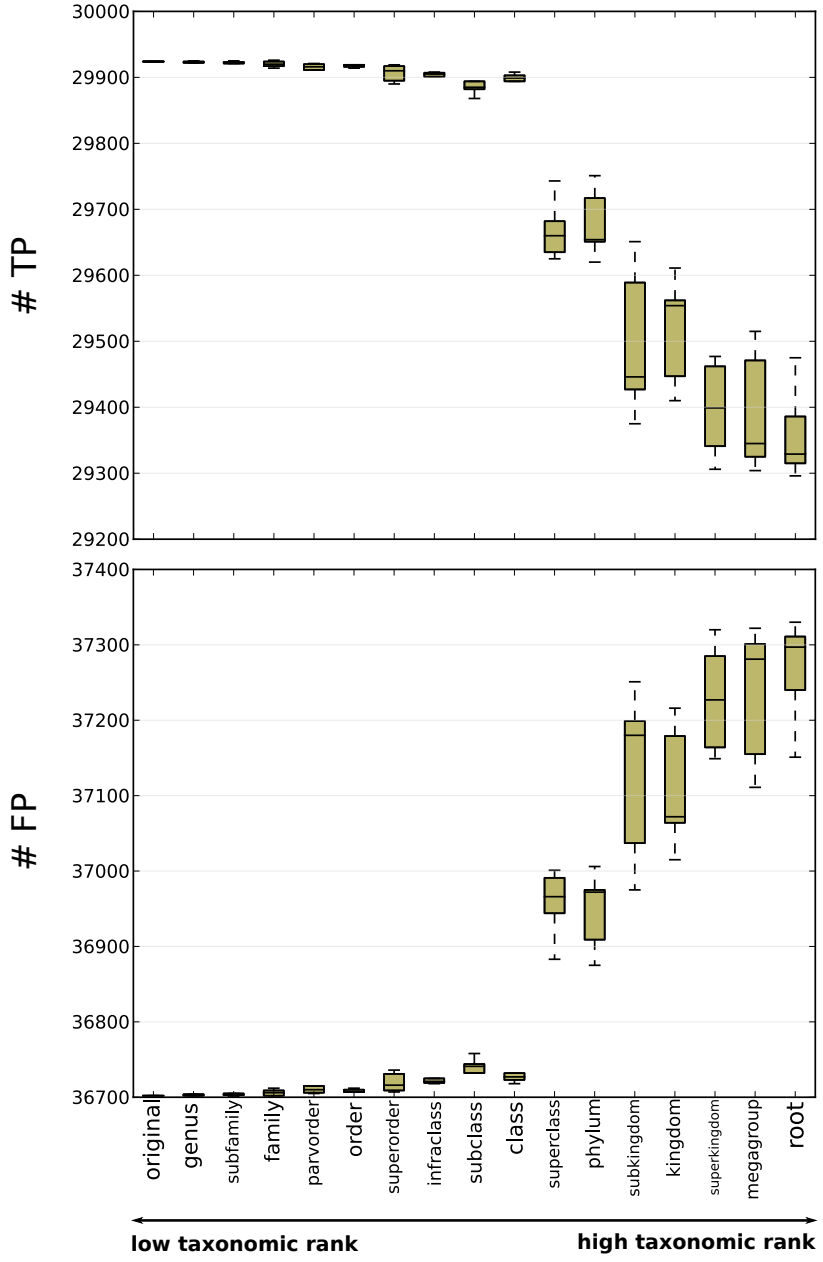



**Figure 4.3:** Shuffling the species tree leaves within taxonomic ranks. The green box plots represent the shuffling for each taxonomic rank repeated five times. On the y axis is the number of True Positives (upper plot) and False Positives (lower plot). The different taxonomic ranks are present in the x axis: from genus (lower) where only the leaves at the the genus taxonomic level are shuffled to the root (higher) where all the leaves of the taxonomic ranks are shuffled.

Note that the number of TP loss and FP gained by shuffling the leaves are not as big as one would expect. This might be explained by TreeHop's basic approach: since it only stops traversing the tree if it finds an ortholog, it can happen that a jump is done across a kingdom and, in this case, shuffling across the taxonomic rank class is not going to influence the jump and the result remains the same as in the un-shuffled case.

This experiment tells us that TreeHop's inference is robust against misspecification of the species tree at lower taxonomic ranks. However, for high taxonomic ranks, shuffling does

make a difference. This is particularly relevant as uncertainty is usually concentrated in lower taxonomic ranks, which we showed does not affect TreeHop's quality.

## 4.3 Different protein types

As already detailed in section 1.3, besides TreeHop components, there are other factors that may influence its performance, for example different types of proteins. We decided to investigate if TreeHop would perform worse on proteins that may present challenges for orthology inference: multi-domain proteins, coiled coil proteins and also large protein families such as the Rab family.

The values of TreeHop's performance on the full dataset and in the different protein classes are shown in Table 4.1. Note that due to the identifier mapping, the set of single and multi-domain proteins does not add up to the full dataset.

In general, there is no major difference in TreeHop's performance among these proteins classes. Except for Rabs, where sensitivity and precision are off by more than 5%. This can be explained by the upper plot which represents BBH alone: for the protein classes where BBH presents the worse performance, TreeHop unleashes its full potential. This suggests that TreeHop is most useful for classes where BBH lacks sensitivity (see Figure 4.4).

| | TreeHop | | | | |
|---|---|---|---|---|---|
| | **No restriction** | **Single domain** | **Multi-domain** | **Coiled coil** | **Rabs** |
| **TP** | 29924 | 8613 | 7596 | 1804 | 174 |
| **FP** | 36702 | 12211 | 8551 | 1786 | 179 |
| **TN** | -29863 | -10282 | -6484 | -1402 | -146 |
| **FN** | -36763 | -10542 | -9663 | -2188 | -207 |
| **SENS** | 0.149 | 0.155 | 0.167 | 0.154 | 0.214 |
| **SPC** | -0.071 | -0.094 | -0.078 | -0.062 | -0.190 |
| **PRC** | -0.109 | -0.117 | -0.102 | -0.091 | -0.137 |
| **ACC** | 0.00008 | -0.009 | 0.007 | 0.01 | 0.015 |

**Table 4.1:** Values of TreeHop performance, according to the different protein classes. The second column corresponds to TreeHop's performance in the full dataset.

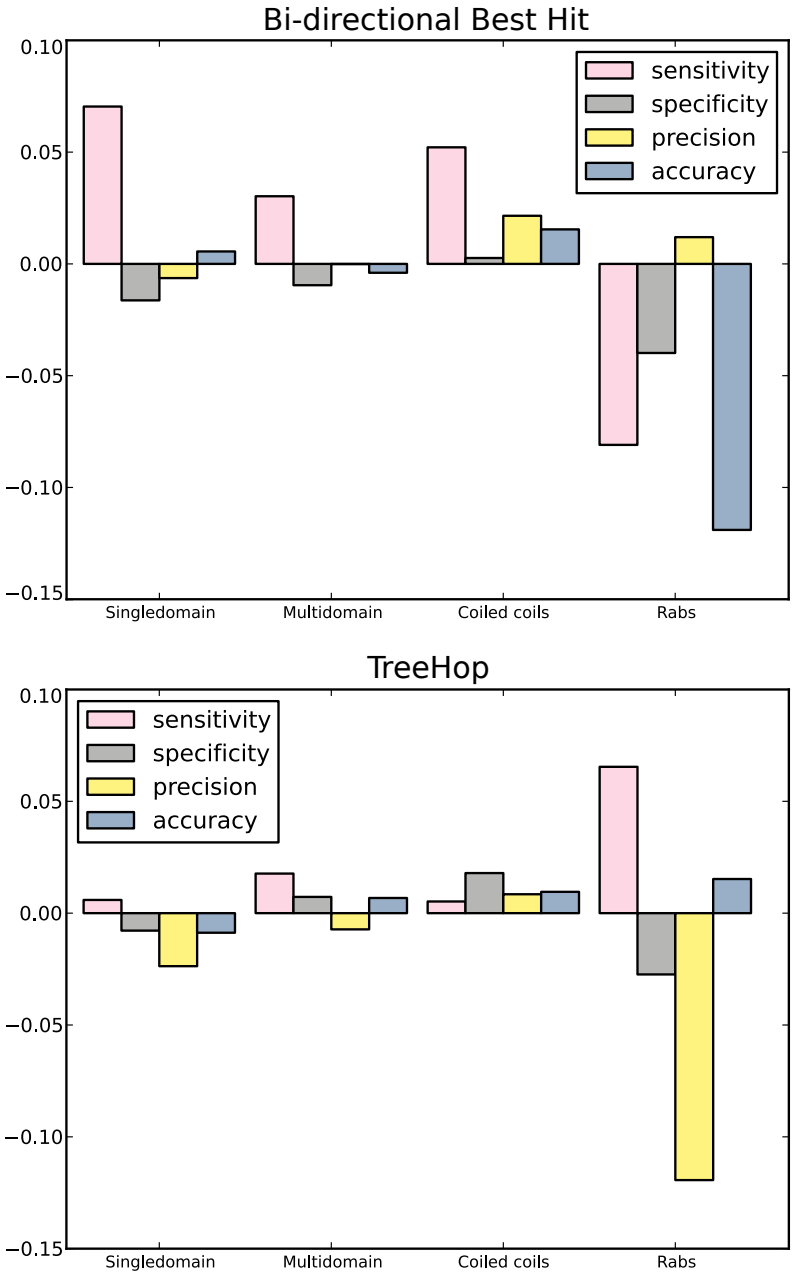



**Figure 4.4:** BBH and TreeHop's performance according to the different protein classes. The values of BBH and TreeHop were subtracted by the values of BBH and TreeHop in the full dataset, respectively.

# 5 TreeHop Optimization

## 5.1 Different method to jump

The default TreeHop implementation uses Bi-directional Best Hit (BBH) as base method and method to jump. Not only the orthologs found by the base method have to be reciprocal best hits, but also TreeHop only assigns an ortholog if it is a reciprocal best hit of the gene it jumps from. As previously mentioned in subsection 1.4.1, BBH does not allow the detection of co-orthologs and it is a very specific method, rather than a sensitive one.

In order to "relax" the jump strategy and observe the effect on TreeHop's performance, we kept the same base method (BBH) and used only the Best Hit (BH) as the mechanism to jump and find orthologs.

In Figure 5.1 we observe a higher number of TP but also a higher number of FP when comparing it to the default TreeHop's implementation (see Figure 3.4). The increased amount of predictions is not surprising given that BH is a superset of BBH: all the bi-directional best hits are also best hits, however, not all best hits are reciprocal. Despite the increase of $\approx 30\%$ in sensitivity, TreeHop's overall performance is worse than before, showing a decrease of $\approx 25\%$ in specificity, $\approx 28\%$ in precision and $\approx 8\%$ in accuracy.

When comparing the values of the different methods to jump (BH vs. BBH), we observe that BH is 2 times more sensitive than BBH. But this comes at a high cost: BH decreases the base method's specificity 4 times more than BBH. BH also decreases the base method's precision approximately 3 times more than BBH, and the increase in accuracy by BBH is 100 times better than using BH. Therefore, BBH is a better choice with respect to the overall performance.

A)

| | BBH | BBH + TreeHop(BH) | TreeHop (BH) |
|---|---|---|---|
| TP | 152949 | 208868 | 55919 |
| FP | 26370 | 159366 | 131950 |
| TN | 480964 | 372310 | -107608 |
| FN | 84233 | 13389 | -70844 |
| SENS | 0.645 | 0.94 | 0.295 |
| SPC | 0.948 | 0.7 | -0.246 |
| PRC | 0.853 | 0.567 | -0.281 |
| ACC | 0.851 | 0.771 | -0.079 |

B)

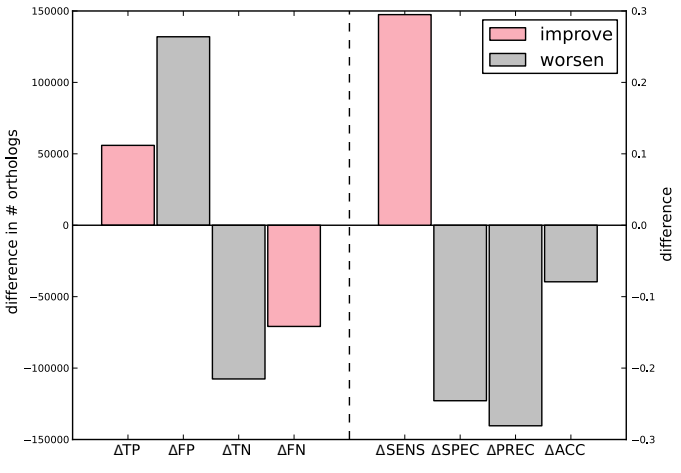



**Figure 5.1:** TreeHop's performance using the Best-Hit approach to jump. **A)** Table with the raw values of the base method (Bi-directional Best hit - BBH), base method + TreeHop using Best-Hit and TreeHop alone. The values of the third column are represented in panel B) for a better visualization of TreeHop's performance. **B)** Graph showing what values TreeHop increases or decreases. It also shows what TreeHop improves (pink bars) and what it diminish(grey bars).

## 5.2 Protein properties

### Protein length

We wanted to investigate whether protein length influences the quality of orthology assignment and if it can be exploited to increase the overall performance of TreeHop. We expect that smaller proteins would more frequently lead to wrong orthology assignment as they are more likely to find sequences with local similarity by chance.

As shown in Figure 5.2, we observe that for smaller proteins the number of FP exceeds the number of TP. For longer proteins the opposite is true: the number of true predictions exceeds the number of false predictions. In order to know where to set the threshold, we computed a ROC curve which shows the relation between specificity and sensitivity for every possible threshold value. Hence, the closer the curve gets to the upper left corner, the more favourable the combination of sensitivity and specificity is.

Despite this two distributions (orthologs correctly assigned by TreeHop and orthologs wrongly assigned by TreeHop according to the protein length) being significantly different ($p-value = 5.42\text{x}10^{-174}$), there is no cut-off (see Figure 5.2 B) which minimizes the number of FP without losing a large amount of true positives.

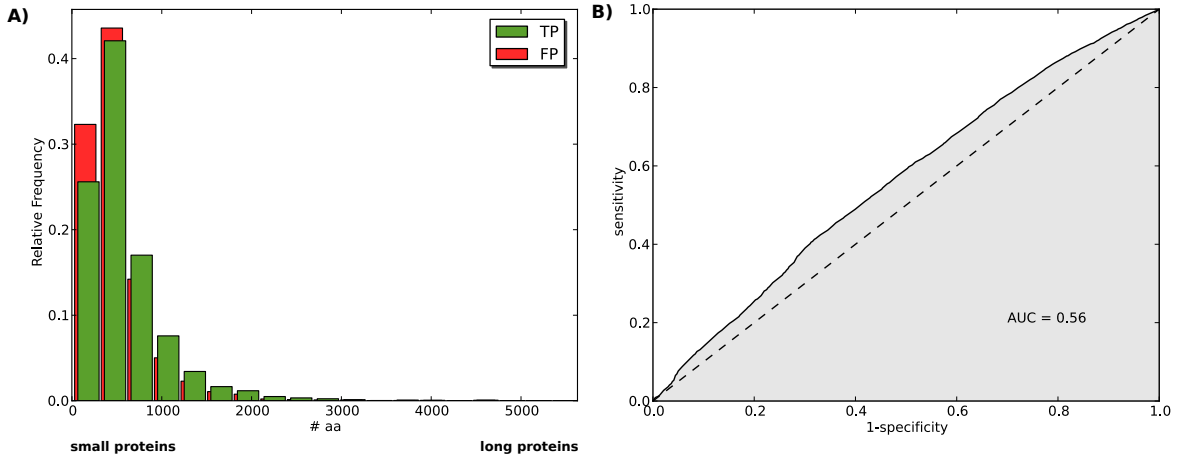



**Figure 5.2:** TreeHop predictions according to their protein length. **A)**True positives (green) and False positives (red) inferred by TreeHop distributed by protein length: smaller proteins on the left and longer proteins on the right of the x axis. The relative frequency in the y axis corresponds to the number of True Positives or False Positives by protein length divided by the total amount of true positives and false positives found by TreeHop, respectively. These two distributions are significantly different ($p - value = 5.42\text{x}10^{-174}$). **B)** ROC curve showing the trade-off between sensitivity and specificity of the previous distributions. The Area Under the Curve (AUC) is approximately 0.5 which means that no clear cut-off can be done to improve TreeHop performance.

## Protein alignment e-value

Similar to the analysis of protein length, we investigated the influence of the e-value between the protein that TreeHop jumped from and the ortholog found on the accuracy of the orthology assignments, as suggested by the sequence divergence analysis shown in Figure 2.2. We expect that higher e-values (within the default threshold) lead to a higher number of FP, whereas smaller e-values result in a higher number of TP. Indeed, in Figure 5.3, we observe that for smaller e-values the number of TP exceeds the number of FP, for higher e-values the opposite is true. As above, in order to find a cut-off to minimize the number of FP while maintaining the number of TP, *i.e.* maximizing sensitivity and specificity, we generated a ROC curve to indicate where to set the best threshold. According to Figure 5.3 B, we set a threshold corresponding to e-value $10^{-9}$. Yet, even at this threshold the gain in accuracy is negligible (data not shown) and we did not further pursue this optimization strategy.

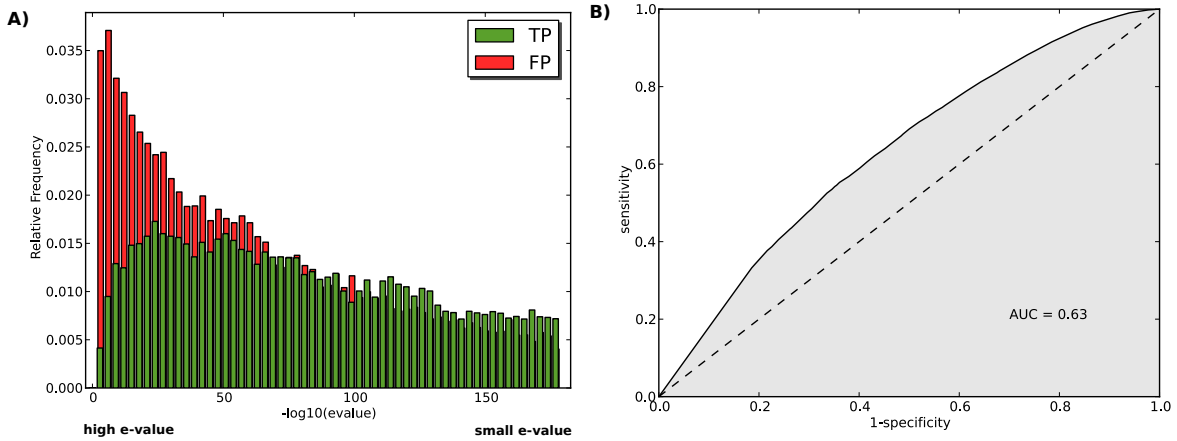



**Figure 5.3:** TreeHop predictions according to the protein e-value. **A)** True positives (green) and False positives (red) inferred by TreeHop distributed by protein e-value: higher e-value on the left and smaller e-value on the right of the x axis. The relative frequency in the y axis corresponds to the number of True Positives or False Positives by e-value divided by the total amount of true positives and false positives found by TreeHop, respectively. These two distributions are significantly different (p-value: 0.0). **B)** ROC curve showing the trade-off between sensitivity and specificity of the previous distributions. The Area Under the Curve (AUC) is approximately 0.63 and shows that to maximize the sensitivity – upper right corner – the cut-off should be done at the e-value $10^-9$ in order to improve TreeHop's performance.

## 5.3 Identification of critical parameters

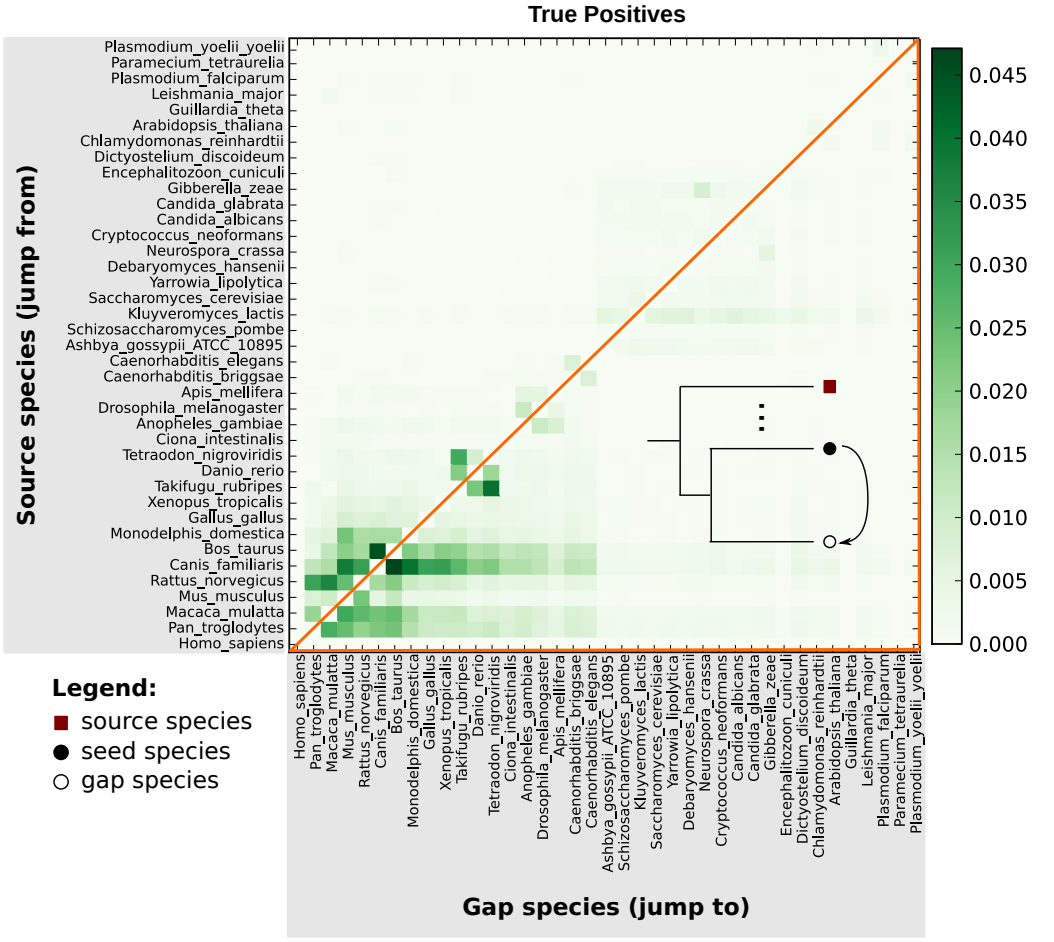
We decided to focus on the analysis of TreeHop's main characteristic: the jumps. Our aim was to find a strategy that could decrease the number of FP while maintaining the number of TP.

### 5.3.1 Relative distance threshold

In the default implementation, TreeHop traverses the species tree and jumps until it finds an ortholog for the current gap. We hypothesized that jumping from species for which the distance to the gap is bigger than the distance between the seed species and the gap, could lead to a gain of more FP. This was based on the fact that closely related species share more orthologs as already shown in Figure 2.1.

The heatmaps of Figure 5.4 represent the amount of orthologs that TreeHop found jumping from a species to another. An evident observation is the fact that the upper triangle has more orthologs detected classified as FP and the lower triangle has more orthologs detected classified as TP. Note that the amount of orthologs is normalized according to the total number of TP and FP in each case. The upper matrix triangle corresponds to the jumps for which the distance between the source species to the gap species is bigger than the distance between the seed species (Human) and the gap species. The lower triangle corresponds to the jumps from which the distance between the source species to the gap species is smaller or equal to the distance from the seed species (Human) to the gap species.

If we designate $d_1$ as the distance between the source species and the gap species and $d_2$ as the distance from the seed species to the gap species, the number of false positives obtained when $d_1$ is bigger than $d_2$ are more than twice the number of true positives (see Figure 5.5). Given this, we implemented this relative threshold, which we subsequently we refer to as stop hopping beyond the seed species. This results in an increase of sensitivity and accuracy of 13% and 0.4%, respectively; and a decrease of 5% in specificity and 8.5% in precision. Hence, given that originally TreeHop improved by 0.008% in accuracy, this corresponds to a 50-fold gain.


True Positives
Source species (jump from)
Gap species (jump to)
Legend:
source species
seed species
gap species
0.045
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Plasmodium_yoelii_yoelii
Paramecium_tetraurelia
Plasmodium_falciparum
Leishmania_major
Guillardia_theta
Arabidopsis_thaliana
Chlamydomonas_reinhardtii
Dictyostelium_discoideum
Encephalitozoon_cuniculi
Gibberella_zeae
Candida_glabrata
Candida_albicans
Cryptococcus_neoformans
Neurospora_crassa
Debaryomyces_hansenii
Yarrowia_lipolytica
Saccharomyces_cerevisiae
Kluyveromyces_lactis
Schizosaccharomyces_pombe
Ashbya_gossypii_ATCC_10895
Caenorhabditis_elegans
Caenorhabditis_briggsae
Apis_mellifera
Drosophila_melanogaster
Anopheles_gambiae
Ciona_intestinalis
Tetraodon_nigroviridis
Danio_rerio
Takifugu_rubripes
Xenopus_tropicalis
Gallus_gallus
Monodelphis_domestica
Bos_taurus
Canis_familiaris
Rattus_norvegicus
Mus_musculus
Macaca_mulatta
Pan_troglodytes
Homo_sapiens

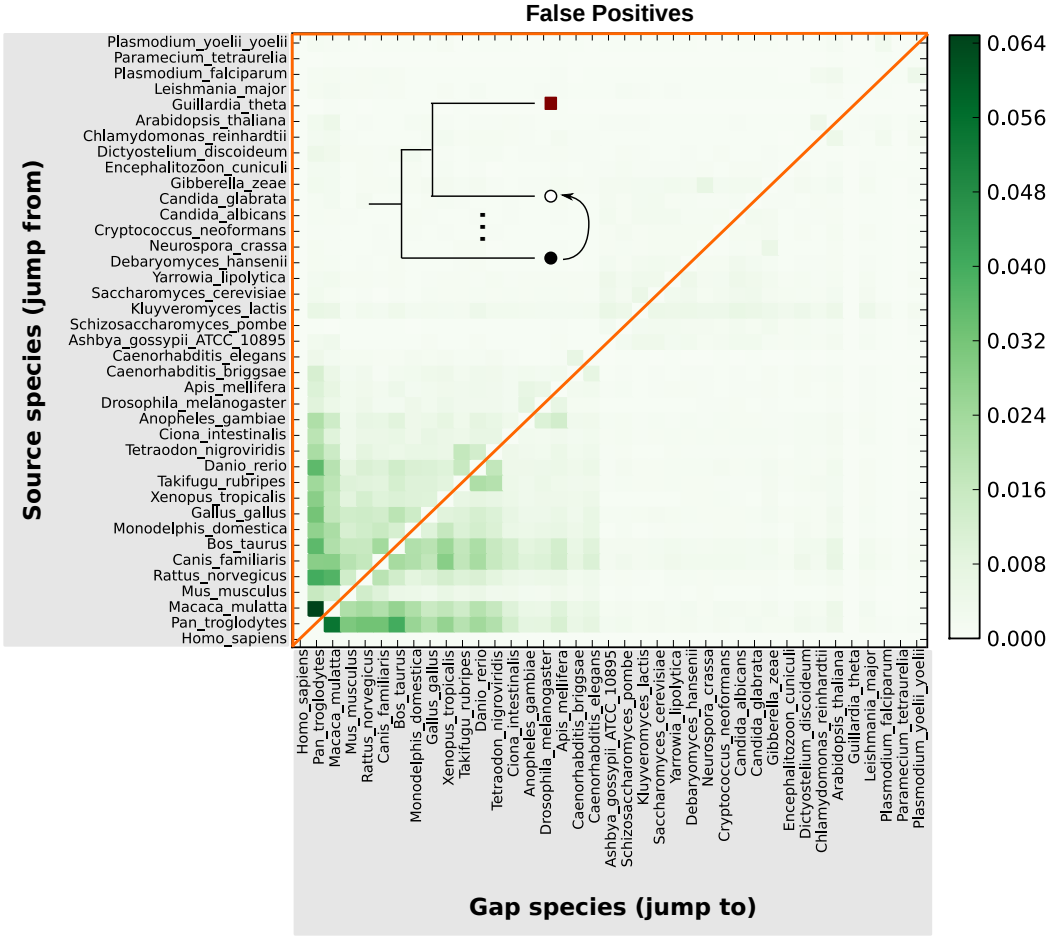

False Positives
Source species (jump from)
Gap species (jump to)
0.064
0.056
0.048
0.040
0.032
0.024
0.016
0.008
0.000
Plasmodium_yoelii_yoelii
Paramecium_tetraurelia
Plasmodium_falciparum
Leishmania_major
Guillardia_theta
Arabidopsis_thaliana
Chlamydomonas_reinhardtii
Dictyostelium_discoideum
Encephalitozoon_cuniculi
Gibberella_zeae
Candida_glabrata
Candida_albicans
Cryptococcus_neoformans
Neurospora_crassa
Debaryomyces_hansenii
Yarrowia_lipolytica
Saccharomyces_cerevisiae
Kluyveromyces_lactis
Schizosaccharomyces_pombe
Ashbya_gossypii_ATCC_10895
Caenorhabditis_elegans
Caenorhabditis_briggsae
Apis_mellifera
Drosophila_melanogaster
Anopheles_gambiae
Ciona_intestinalis
Tetraodon_nigroviridis
Danio_rerio
Takifugu_rubripes
Xenopus_tropicalis
Gallus_gallus
Monodelphis_domestica
Bos_taurus
Canis_familiaris
Rattus_norvegicus
Mus_musculus
Macaca_mulatta
Pan_troglodytes
Homo_sapiens

**Figure 5.4** ***(preceding page)*:** Fraction of TP and FP when jumping from the source species to the gap species. In the heatmap correspondent to True Positives, a darker green pattern is more concentrated in the lower triangle, whereas the in the heatmap correspondent to False Positives, the same is observed but in the upper triangle. Note that the lower triangle corresponds to jumps between species for which the distance is smaller than the seed gene (red square) to the gap species (white circle), this is represented by the inside cartoon. And the upper triangle corresponds to jumps between species for which the distance is bigger than the seed gene (red square) to the gap species (white circle), this is represented by the inside cartoon.

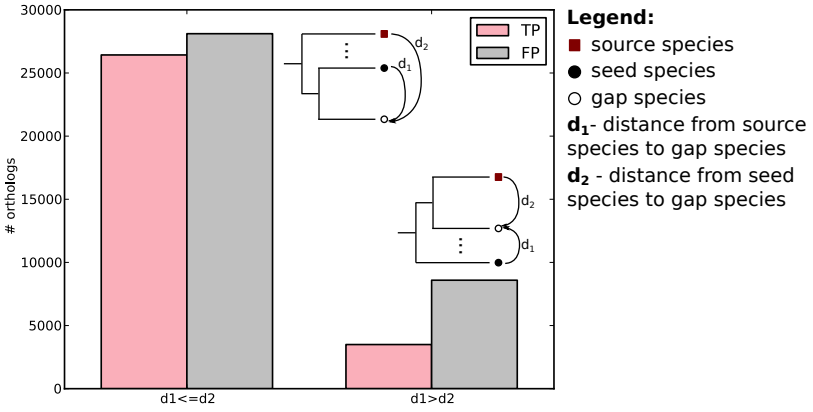
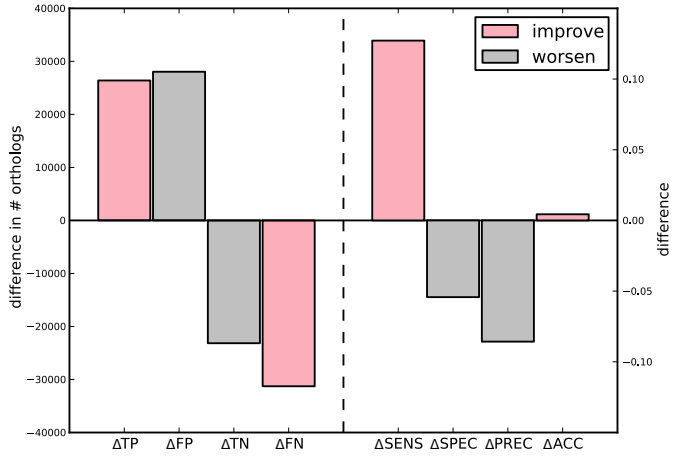



**Figure 5.5:** Number of True Positives and False positives according to the relation between the distance from the seed species to the gap $d_1$ and distance from the source species to the gap $d_2$.

**A)**




**B)**

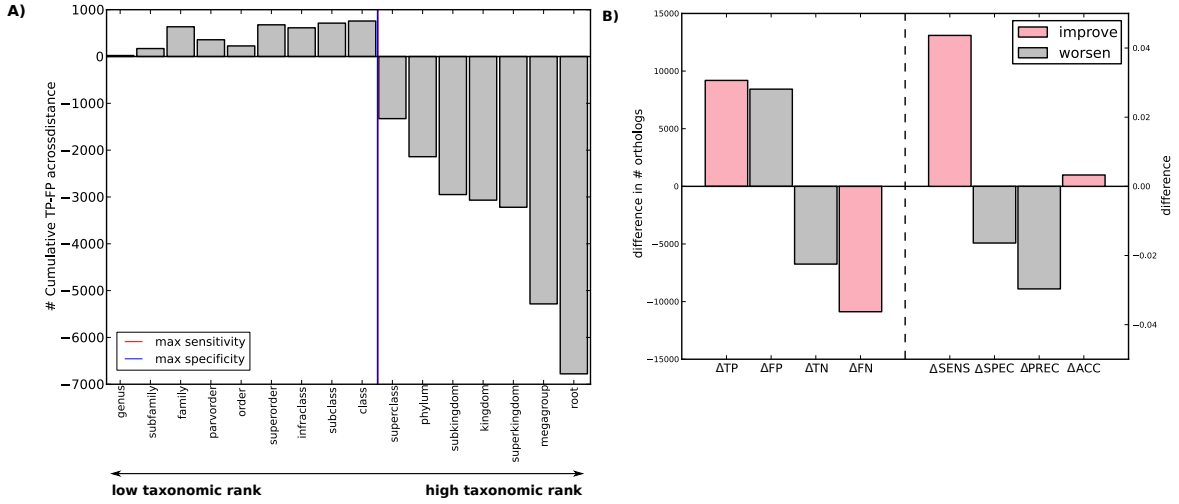
| | TreeHop | | | | |
|---|---|---|---|---|---|
| | **No restriction** | **stop hopping beyond seed** | | **No restriction** | **stop hopping beyond seed** |
| **TP** | 29924 | 26395 | **SENS** | 0.149 | 0.127 |
| **FP** | 36702 | 28043 | **SPC** | -0.071 | -0.054 |
| **TN** | -29863 | -23160 | **PRC** | -0.109 | -0.086 |
| **FN** | -36763 | -31278 | **ACC** | 0.00008 | 0.004 |

**Figure 5.6:** TreeHop performance after implementing a relative threshold – stop hopping beyond the seed species. **A)** Graph showing what TreeHop improves (pink bars) and what it diminish (grey bars). **B)** Table showing the previous values obtained by the default algorithm and the values after implementing the relative threshold.

### 5.3.2 Absolute distance threshold

In a similar manner, we explored how TreeHop performs when jumping within each taxonomic rank. We expected that jumps within a lower taxonomic rank would probably result in higher number of correct predictions, whether jumps within higher taxonomic ranks would probably lead to a higher number of wrong orthology predictions. To investigate this, we counted the number of TP and FP for each jump within a taxonomic rank. Note that the number of orthologs (TP and FP) in each rank are the sum of all the values in the lower ranks. For example, if we implement a threshold to only jump between the taxonomic rank "class" this means that jumps between the taxonomic rank "family" would also be done.

Figure 5.7 shows the amount of TP subtracted by the number of FP for jumps within the different taxonomic ranks. If we want to maximize the specificity while still increasing the accuracy of the algorithm, we should impose a cut-off at the taxonomic rank corresponding to the highest positive bar. In the case of maximizing sensitivity while still increasing accuracy the cut-off should be at the taxonomic rank corresponding to the last positive bar. In this case, both cut-offs coincide at the taxonomic rank "class". We applied this cut-off to TreeHop and resulted in an increase of $\approx 4\%$ in sensitivity and 0.3% in accuracy and a decrease of $\approx 2\%$ in specificity and 3% in precision (see Figure 5.7 B and C). The number of TP found with this strategy is small, however it increases over 35% the accuracy when comparing with the values from the default implementation.




C)

| | TreeHop | | | | |
|---|---|---|---|---|---|
| | No restriction | taxonomic rank class | | No restriction | taxonomic rank class |
| TP | 29924 | 9191 | SENS | 0.149 | 0.044 |
| FP | 36702 | 8433 | SPC | -0.071 | -0.016 |
| TN | -29863 | -6746 | PRC | -0.109 | -0.03 |
| FN | -36763 | -10878 | ACC | 0.00008 | 0.003 |

**Figure 5.7:** TreeHop performance by taxonomic rank. **A)** The difference between True Positives and False Positives along the different taxonomic ranks. The red line indicates where to make a cut-off if we want to maximize sensitivity while still increasing the accuracy. The blue line indicates where to make a cut-off if we want to maximize specificity while still increasing the accuracy. In this case, both lines coincide at the taxonomic rank *class*. **B)** TreeHop performance after applying the cut-off at the taxonomic rank *class*. It shows what values TreeHop improves (pink bars) and what it diminish (grey bars). **C)** Table showing the previous values regarding the default algorithm and the values after implementing the absolute threshold.

### 5.3.3 Hop consistency

TreeHop jumps from the closest species with respect to the gap it is trying to fill, but in some cases is more than one closest species (see Figure 5.8). The default implementation of TreeHop jumps from one of the species randomly and if it does not find an ortholog it jumps from the next species. We decided to consider all the jumps from species at the same distance to the gap to increase the confidence in the predictions which may result in a better overall performance.

For the orthologs assigned by the base method which were all at the same distance from a certain gap, we considered three different ways to evaluate the jumps: i) choose two of the species and if they agree on the same ortholog, that is considered a hit, if they do not agree no ortholog is assigned to the gap; ii) half or more of the orthologs predicted from the different source species have to be the same; iii) the ortholog predicted by the majority of the species is assigned. An example of how the different criteria work are shown in Figure 5.8 B, C and D, respectively.

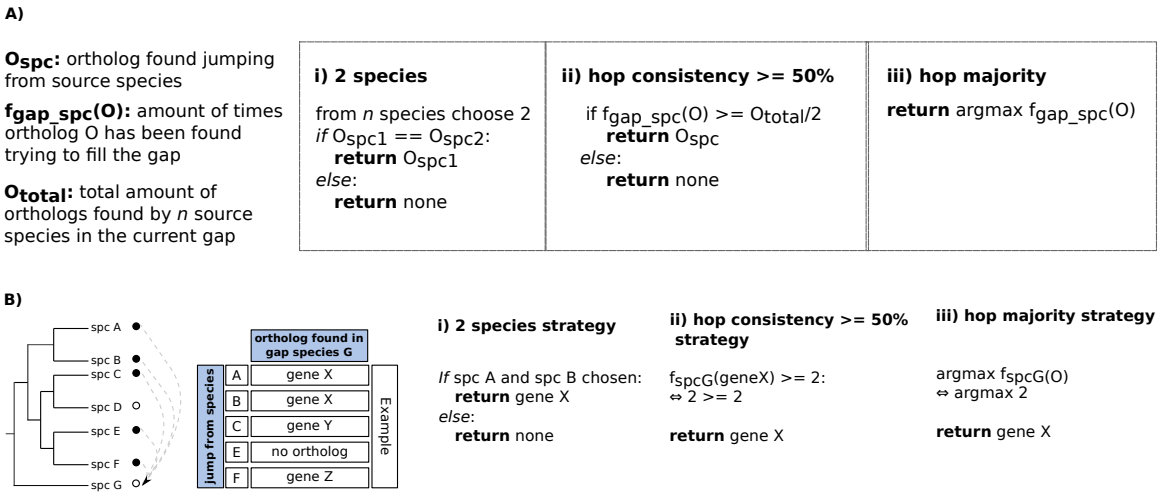



**Figure 5.8:** Hop consistency. **A)** Schematic example of the three different criteria used to test for hop consistency. **B)** Example of a scenario to apply hop consistency. On the tree on the left, all species are at the same distance to species G. The table next to it exemplifies possible outputs of the jumps from the different species to the gap (species G). On the right is shown the output of TreeHop using the three different criteria. For the strategy i) the output could be gene X or no gene, and for the other strategies the output would be gene X. Note that there is no jump from species D, as it is a gap. And the output from the jump between species E to species G is no ortholog, so this is not further considered when applying the different hop consistency strategies.

Table 5.1 shows TreeHop's performance according to the different hop consistency strategies. If we compare all the approaches between each other, we notice that the highest increase of accuracy (0.4%) and the least decrease in specificity (-0.9%) and precision ($\approx$ 2%) is obtained with the 2 species approach, while the highest gain in sensitivity ($\approx$ 5%) is obtained both with $\geq$50% consistency and hop majority.

| | TreeHop | | | |
|---|---|---|---|---|
| | **No restriction** | **2 species** | **≥50%** | **Hop majority** |
| **TP** | 29924 | 6958 | 10292 | 10306 |
| **FP** | 36702 | 4828 | 9961 | 10002 |
| **TN** | -29863 | -3952 | -8032 | -8049 |
| **FN** | -36763 | -7834 | -12221 | -12259 |
| **SENS** | 0.149 | 0.032 | 0.049 | 0.049 |
| **SPC** | -0.071 | -0.009 | -0.019 | -0.0194 |
| **PRC** | -0.109 | -0.016 | -0.035 | -0.035 |
| **ACC** | 0.00008 | 0.004 | 0.003 | 0.003 |

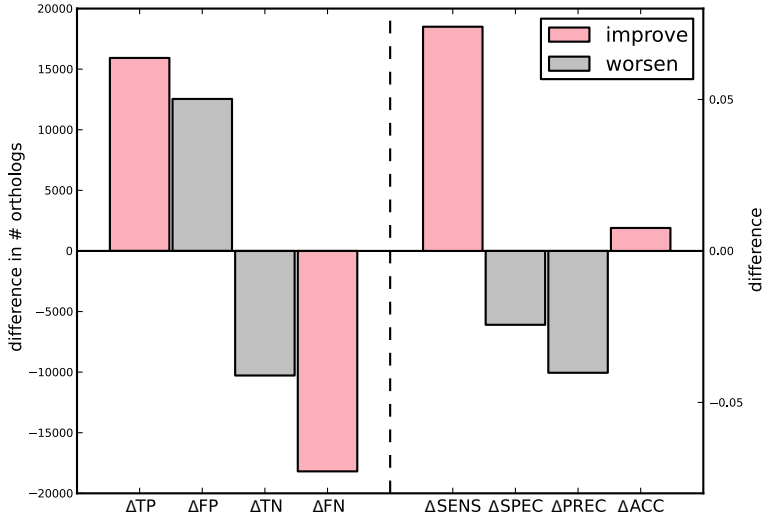
**Table 5.1:** TreeHop's performance according to the different hop consistency strategies.

### 5.3.4 First hit

Another possible approach to improve TreeHop's performance is to only consider the result of the jump from the closest species (which contains an ortholog assigned by the base method). If the jump from the closest species to the current gap is successful, we consider it a hit, if not, no ortholog is assigned and we do not continue traversing the tree. Despite a decrease in TP, the number of FP for this strategy decreased more than half, which led to an increase of 0.8% of accuracy and $\approx$ 7% of sensitivity and a decrease of $\approx$ 2% and 4% in specificity and precision, respectively (see Figure 5.9).

This results tell us that it is more likely to find a TP jumping from closest species and that is more likely to find FP jumping from distantly related species.

**A)**




**B)**

| | TreeHop | | | | |
|---|---|---|---|---|---|
| | **No restriction** | **First Hit** | | **No restriction** | **First Hit** |
| **TP** | 29924 | 15925 | **SENS** | 0.149 | 0.074 |
| **FP** | 36702 | 12542 | **SPC** | -0.071 | -0.024 |
| **TN** | -29863 | -10278 | **PRC** | -0.109 | -0.04 |
| **FN** | -36763 | -18189 | **ACC** | 0.00008 | 0.008 |

**Figure 5.9:** TreeHop performance after applying the First Hit approach. **A)** It shows what TreeHop improves (pink bars) and what it diminish (grey bars). **B)** Table showing the previous values obtained by the default algorithm and the values after implementing the first hit cut-off.

## 5.4 Overall optimization

So far we tested different approaches to improve TreeHop's performance, however, these were tested independently. In this section we combined all the previous strategies in order to understand which one provides the best TreeHop's overall performance.

The previous results for each strategy show that increasing sensitivity is never possible without decreasing specificity. How do we evaluate if the loss in specificity is balanced by the gain in sensitivity? That depends on the purpose of TreeHop, *i.e.* what relative importance we give to sensitivity and specificity. Here, we distinguished three scenarios: same importance to sensitivity and specificity (Figure 5.10), more importance to sensitivity than specificity (Figure 5.11 A) and more importance to specificity than to sensitivity (Figure 5.11 B).

In order to visualize these different weights, we plotted lines corresponding to an unchanged (weighted) sensitivity / specificity ratio. If we want to either give the same weight to sensitivity and specificity or more importance to sensitivity, then the approach to choose is the TreeHop's default implementation, *i.e.* TreeHop with no other parameter added. If we want to give more weight to specificity than to sensitivity then we should choose TreeHop plus the First Hit strategy. Note that any parameter or combination is better than the base method (Bi-directional Best Hit) alone.

The values of sensitivity and specificity reflect a combination of positives (TP and FN) and negatives(TN and FP), *i.e.* they do not reflect directly the absolute numbers of these values. In case we are interested in the best trade-off between True Positives and False Positives, then TreeHop plus the parameter stop hopping beyond the seed species gives the best result (see Figure 5.12).

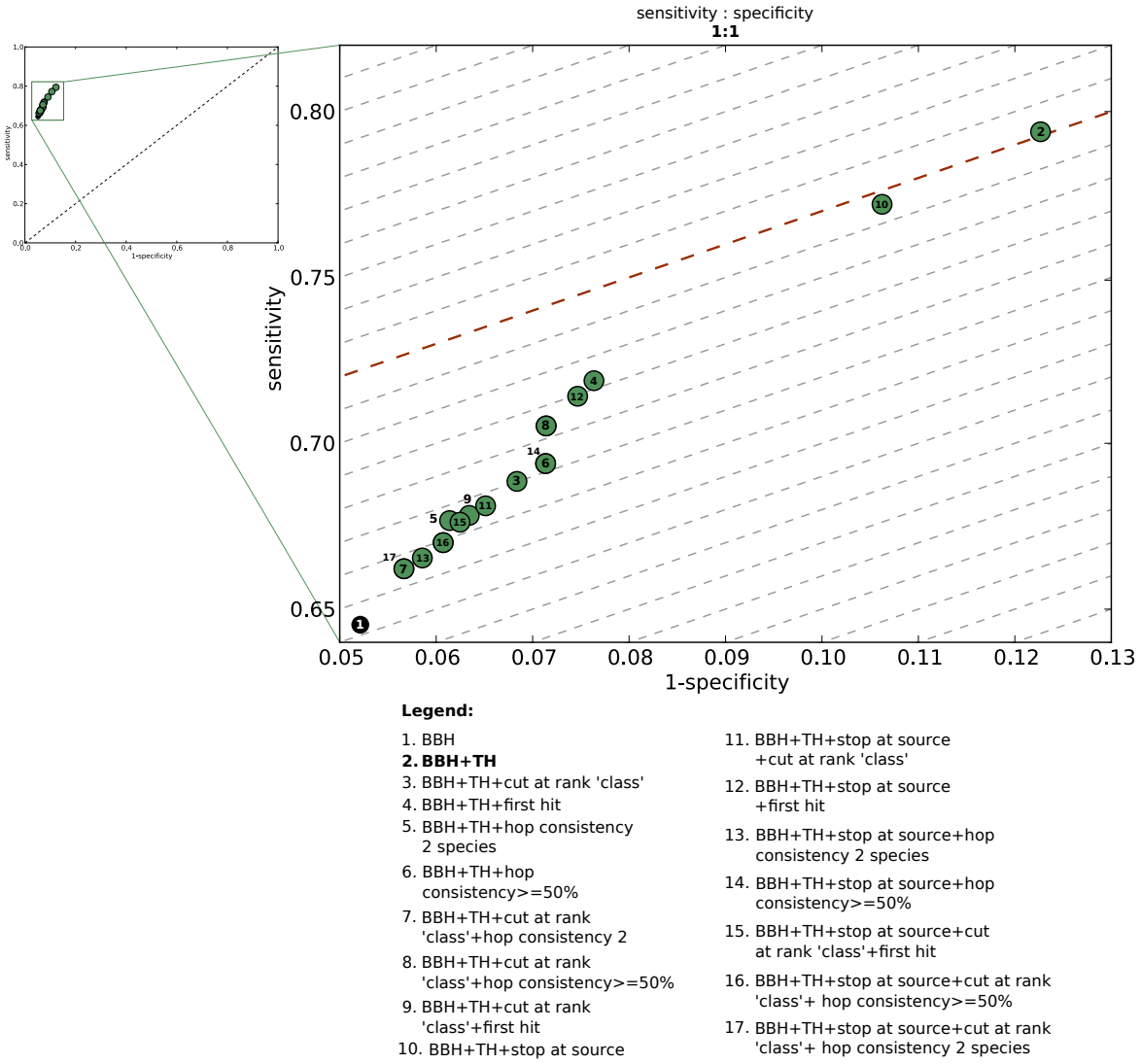



**Figure 5.10:** Combination of all the different parameters. The green solid circles correspond to each of the TreeHop's parameters and their combinations. The black solid circle represents the base method (Bi-directional Best Hit) alone. The grey lines in the background indicate the slope which symbolizes the weight that we give to sensitivity and specificity. In this case, we give the same importance to both. Since the best performance is on the top left of the graph, from there we observe that the first line (red line) touches the green circle number 2. This indicates that giving the same importance to both sensitivity and specificity one should choose TreeHop with no other parameter.

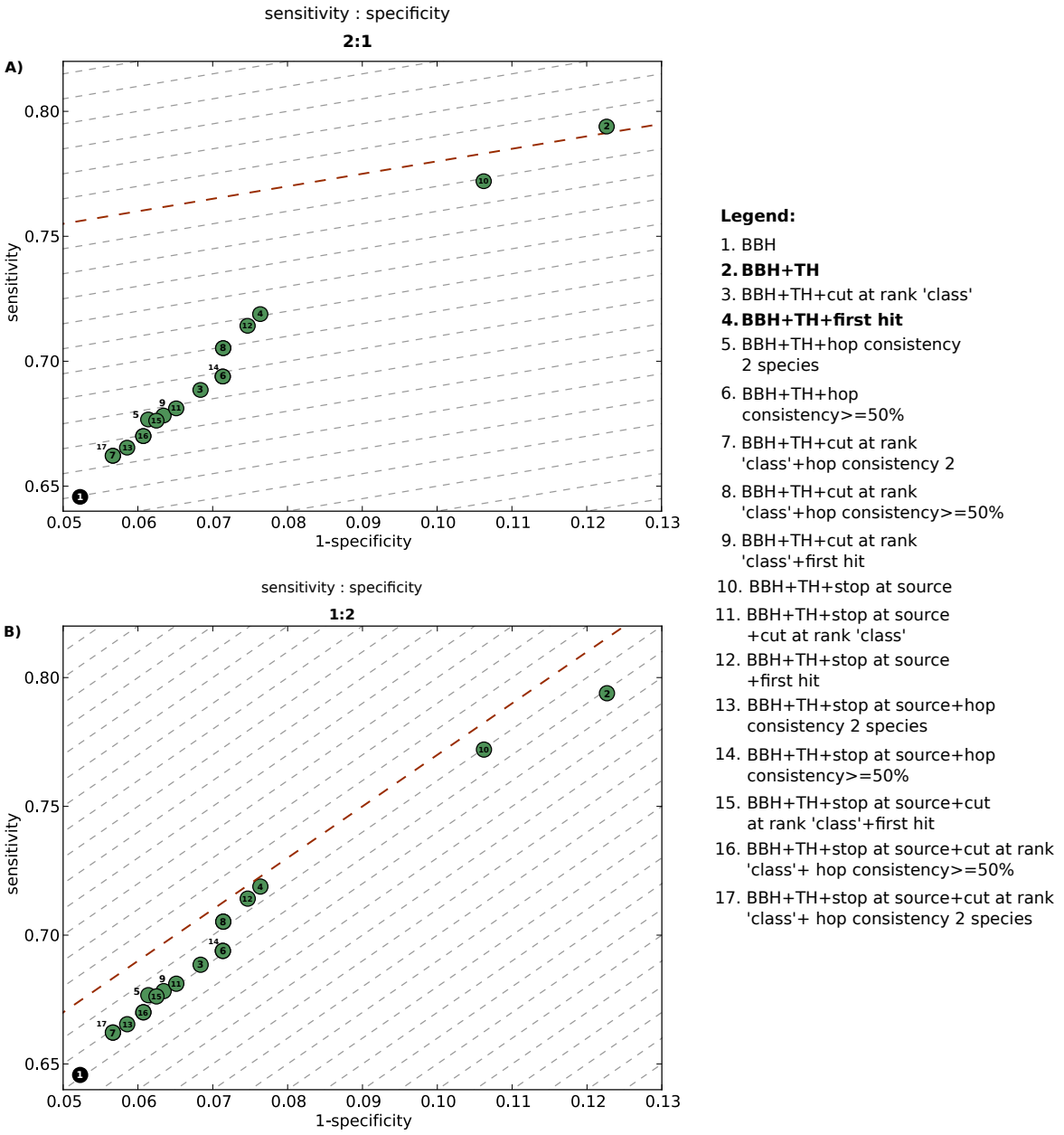



**Figure 5.11:** Combination of all the different parameters giving different weights to sensitivity and specificity. The green solid circles correspond to each of the TreeHop's parameters and their combinations. The black solid circle represents the base method (Bi-directional Best Hit) alone. The grey lines in the background indicate the slope which symbolizes the weight that we give to sensitivity and specificity. In **A)** the slope corresponds to 2:1 which means two sensitive units per one specificity unit. The approach to choose in case we want to give more importance to sensitivity than to specificity is TreeHop with no other parameter. In **B)** the slope corresponds to 1:2 which means one sensitive unit per two specificity units. The approach to choose in case we want to give more importance to specificity than to sensitivity is TreeHop plus First Hit strategy.

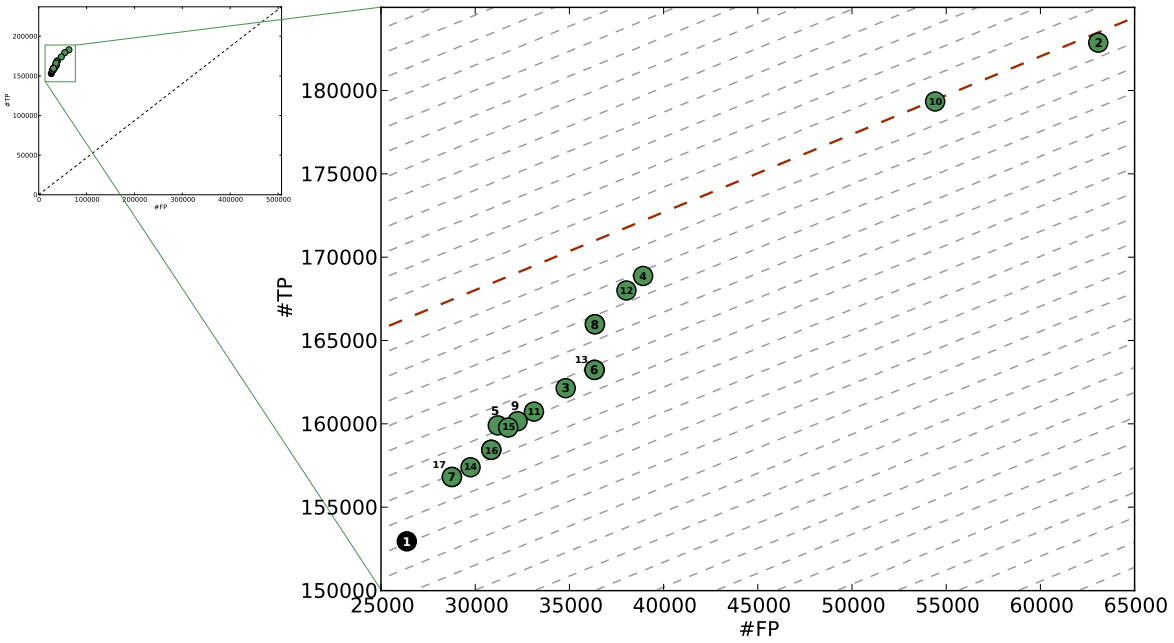



**Figure 5.12:** Combination of all the different parameters. The green solid circles correspond to each of the TreeHop's parameters and their combinations. The black solid circle represents the base method (Bi-directional Best Hit) alone. The combination with the best trade-off is TreeHop plus the parameter stop hopping beyond the seed species.

# 6 Conclusion & Future Perspectives

## Conclusion

The detection of orthology relationships between genes is very important. Two main reasons are the evolutionary history reconstruction and functional annotation transfer. However, the attempt to re-construct events that occurred millions of years ago is a difficult task. A large amount of methods has been developed to improve orthology inference by combining phylogeny, biochemical structures, gene position and function. Despite this variety of different strategies, none does satisfactorily solve the problem, because all of them face the classical trade-off: an efficient method is not the most accurate (graph-based methods), and the most accurate methods lack efficiency (tree-based methods).

Here we propose an algorithm that improves the performance of Bi-directional Best Hits in an efficient manner. While on one hand tree-based orthology inference may require the use of large computer clusters even for a moderate number of taxa[1], all vs. all BBH does not allow to restrict the search for orthologs only to the proteins of interest. The 'one vs. all' approach of TreeHop improves on both of these shortcomings.

We tried several strategies to optimize the algorithm performance and we observe that we cannot escape the classical trade-off between sensitivity and specificity. In addition, the parameters that provide the best ratio TP/FP are the ones which are more stringent, resulting in a small number of additional orthologous assignments. From this it became clear that the decision of implementing a certain strategy, depends on what we want the algorithm to be. This is the reason we offer a spectrum of different usage possibilities: TreeHop, TreeHop + stop hopping beyond the seed species, TreeHop + stop hopping beyond the seed species + taxonomic rank threshold and TreeHop + first hit (see Figure 6.1). The choice of these different strategies depends if we want TreeHop to be more specific or more sensitive. Note that independently of this choice, TreeHop always increases the sensitivity and accuracy of the base method.

[1] `sco.h-its.org/exelixis/countManual7.0.4.php`

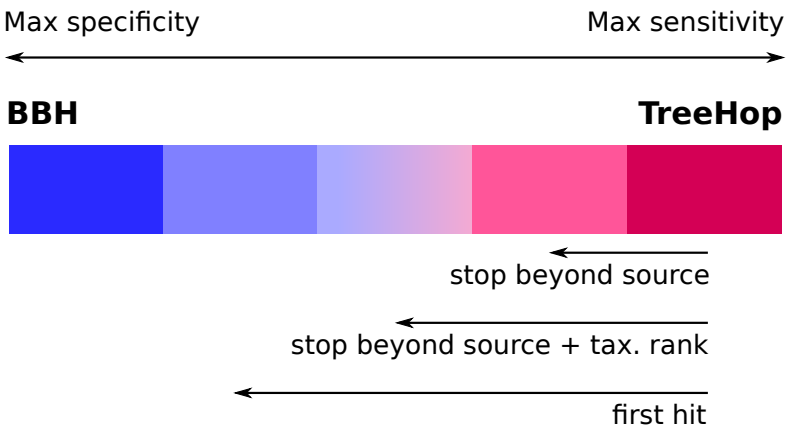



**Figure 6.1:** Schema summarizing the different strategies that could be implemented in TreeHop algorithm.

## Future Perspectives

The algorithm presented in this thesis was implemented using an efficient methology for the search of orthologs, Bi-directional Best Hit, which despite of its simplicity, is still one of the most used strategies. However, and as a consequence of using this approach as a base-method and jump strategy, our algorithm can only provide with one-to-one orthology relationships. This can lead to an incomplete orthology assignment given that we are not considering the in-paralogous genes (co-orthologs of the gene of interest). Given that the algorithm is prepared to receive as input any orthology detection profile, we could use the method Inparanoid (Ostlund et al., 2010), an extension of the BBH method, which finds in-paralogy relationships. Moreover, if there are two or more very similar sequences in the same genome this can lead to a miss-assignment. For example, if gene $x$ and gene $y$ from a genome A are very similar, and we *blast* gene $x$ against a genome B, the reciprocal hit might result in gene $y$. Since this is not considered by definition a bi-directional best hit, this leads to a miss assignment of an ortholog. To overcome this we could consider more than one hit in the case of having the same e-value in the result sequences.

Furthermore, our main purpose is to make TreeHop useful to others. In particular to biologists, which may not be familiar with Python, a possibility to provide access to TreeHop maybe via a web-tool. The advantage is its user-friendliness and the disadvantage is the less flexibility for the user which would be limited by the species tree and genomes available in the web-tool. Another possibility is to create a Python package which would be less intuitive for the user, but would give them the possibility to choose their genomes of interest and species tree.

# 7 Material & Methods

## Database

We constructed a database using PostgreSQL version 8.4 to store the proteomes and the results of orthology detection. The database diagram is illustrated in

## Algorithm implementation

The algorithm was implemented using the Python programming language version 2.6.

### Bi-directional Best Hit

The Bi-directional Best Hit method relies on BLAST as the underlying homology detection tool. As BLAST is a local alignment algorithm, high-scoring matches between parts of proteins, such as conserved domains, may receive high scores even though they do not reflect a common origin for the proteins as a whole. To avoid drawing conclusions from fragment matches of this type, BLAST homology inference is only accepted if the region aligned by BLAST corresponds to a large enough fraction of the lengths of the proteins. Only sequences that aligned with a continuous region longer than 50% of the query sequence and with a significant similarity (e-value $< 10^{-3}$) were selected.

### Tree construction

The species tree was constructed based on NCBI taxonomy (see Figure 7.2 and Figure 7.3). The Taxonomy Browser contains a database of all organisms represented in the NCBI sequence database, and can automatically build a species tree using species selected by the user. The output format was newick.

The taxonomic ranks' classification were obtained from the NCBI taxonomy. For the taxonomic ranks without classification, a the higher classified taxonomic rank was assigned.

### Tree traversing

To traverse the species tree we used the Pyhton package ETE2. ETE2 is a python programming toolkit that assists in the automated manipulation, analysis and visualization of phylogenetic and other type of trees. It provides a wide range of tree handling methods, node annotation features, programmatic access to the PhylomeDB database, and automatic orthology and paralogy prediction methods.

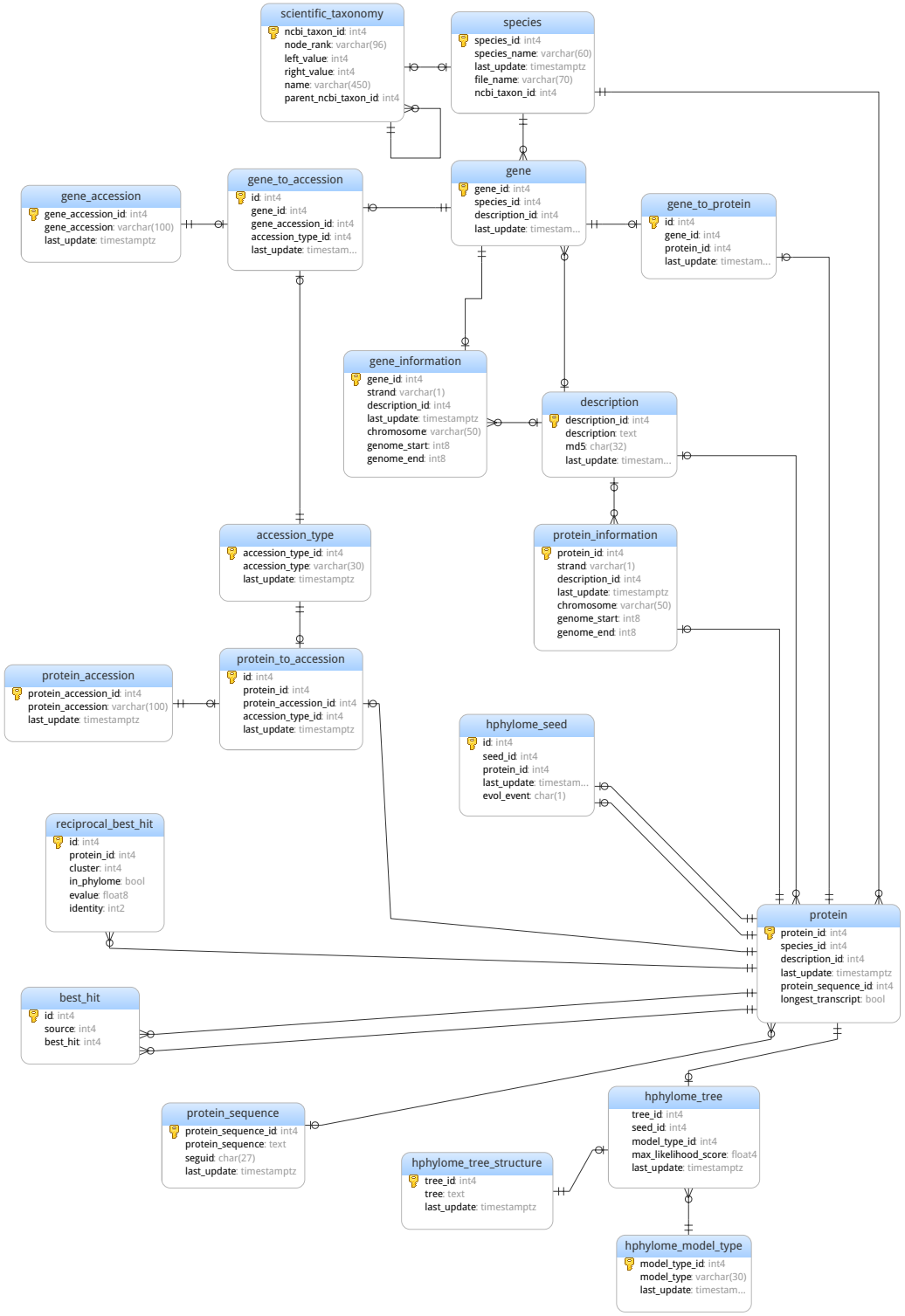



**Figure 7.1:** Database diagram.

# Validation

## PhylomeDB

PhylomeDB is a public database for complete collections of gene phylogenies (phylomes). It allows users to interactively explore the evolutionary history of genes through the visualization of phylogenetic trees and multiple sequence alignments. The automated pipeline used to reconstruct trees aims at providing a high-quality phylogenetic analysis of different genomes, including Maximum Likelihood or Bayesian tree inference, alignment trimming and evolutionary model testing.

### Dataset

Proteomes derived from 39 fully sequenced eukaryotic genomes used to construct the Human Phylome were downloaded from the PhylomeDB (http://phylomedb.org/phylome_1; ftp://phylomedb.org/phylomedb/proteomes/) which corresponds to Ensembl v36 and the Integr8 database at EBI, except those of *Candida albicans*, *N. crassa*, and *C. reinhardtii*.

### Homologs

The trees for each Human protein were downloaded from the PhylomeDB. Only the file containing the best scoring trees were considered. We uploaded the tree files to our database to increase the homology search efficiency. In order to only compare our detection results with the orthologous genes present in the gene trees (and not paralogous genes), we used the function get_evol_events from the ete2 package which provides us with the information about the evolutionary event present in each internal node (S – speciation event, D – duplication event).

## Statistics

The values of sensitivity, specificity, precision and accuracy were obtained by the combination of the comfusion matrix values: True Positive (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN).

$\text{Sensitivity} = \frac{TP}{TP+FN}$; $\text{Specificity} = \frac{TN}{TN+FP}$; $\text{Precision} = \frac{TP}{TP+FP}$; $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$

The values of TP, FP, TN, FN, sensitivity, specificity, precision and accuracy of TreeHop alone, *i.e.* without the base method output were obtained using these formulas: $\Delta\text{TP} = \text{TP}_{\text{base method+TreeHop}} - \text{TP}_{\text{base method}}$; $\Delta\text{FP} = \text{FP}_{\text{base method+TreeHop}} - \text{FP}_{\text{base method}}$; $\Delta\text{TN} = \text{TN}_{\text{base method+TreeHop}} - \text{TN}_{\text{base method}}$; $\Delta\text{FN} = \text{FN}_{\text{base method+TreeHop}} - \text{FN}_{\text{base method}}$; $\Delta\text{SENS} = \text{Sensitivity}_{\text{base method+TreeHop}} - \text{Sensitivity}_{\text{base method}}$; $\Delta\text{SPC} = \text{Specificity}_{\text{base method+TreeHop}} - \text{Specificity}_{\text{base method}}$; $\Delta\text{PRC} = \text{Precision}_{\text{base method+TreeHop}} - \text{Precision}_{\text{base method}}$; $\Delta\text{ACC} = \text{Accuracy}_{\text{base method+TreeHop}} - \text{Accuracy}_{\text{base method}}$.

### 70 manually curated families

#### Dataset

The proteomes of 12 species (*Caenorhabditis elegans*, *Drosophila melanogaster*, *Ciona intestinalis*, *Danio rerio*, *Tetraodon nigroviridis*, *Gallus gallus*, *Monodelphis domestica*, *Mus musculus*, *Rattus norvegicus*, *Canis familiaris*, *Pan troglodytes*, *Homo sapiens* ) were taken from ensemblv60 [1] (Trachana et al., 2011b).

#### Homologs

The homology seeds were from 3 different species: human, fly and zebrafish. 67 out of 70 RefOGs (Reference Orthologous Groups) have human sequences in the species tree, 1 RefOG (RefOG013) only has fly and the other two (RefOG015, RefOG010) were generated from zebrafish. To be consistent in the analysis of the algorithm performance we decided to only consider the 67 RefOGs with the human sequences and use them as seeds in our method.

#### Different categories

#### Rates of evolution

The RefOGs were classified according to different rates of evolution based on the MeanID score (derived from MSA of each family). There are fast-evolving (MeanID<0.5), medium-evolving (0.7>MeanID>0.5), slow-evolving (MeanID>0.7) RefOGs.

#### Protein family size

The RefOGs were separated into large (more than 40 genes), medium (between 14 and 40 genes) and small (less than 14 genes).

#### Alignment quality

The RefOGs we classified based on the quality of the alignment,according to their NorMD score : high-quality (norMD>0.6) and low-quality (norMD<0.6).

### Statistics

The values were calculated as in Figure 7.

## Protein classes

The protein set of Rab GTPases were obtained from the Rabifier annotation (Diekmann et al., 2011). The coiled coil protein set were obtained using PAIRCOIL2 (McDonnell et al., 2006). The multi-domain protein set was obtained from the Superfamily database, where two or more domains were annotated (Gough et al., 2001).

[1] `http://nov2010.archive.ensembl.org/info/data/ftp/index.html`

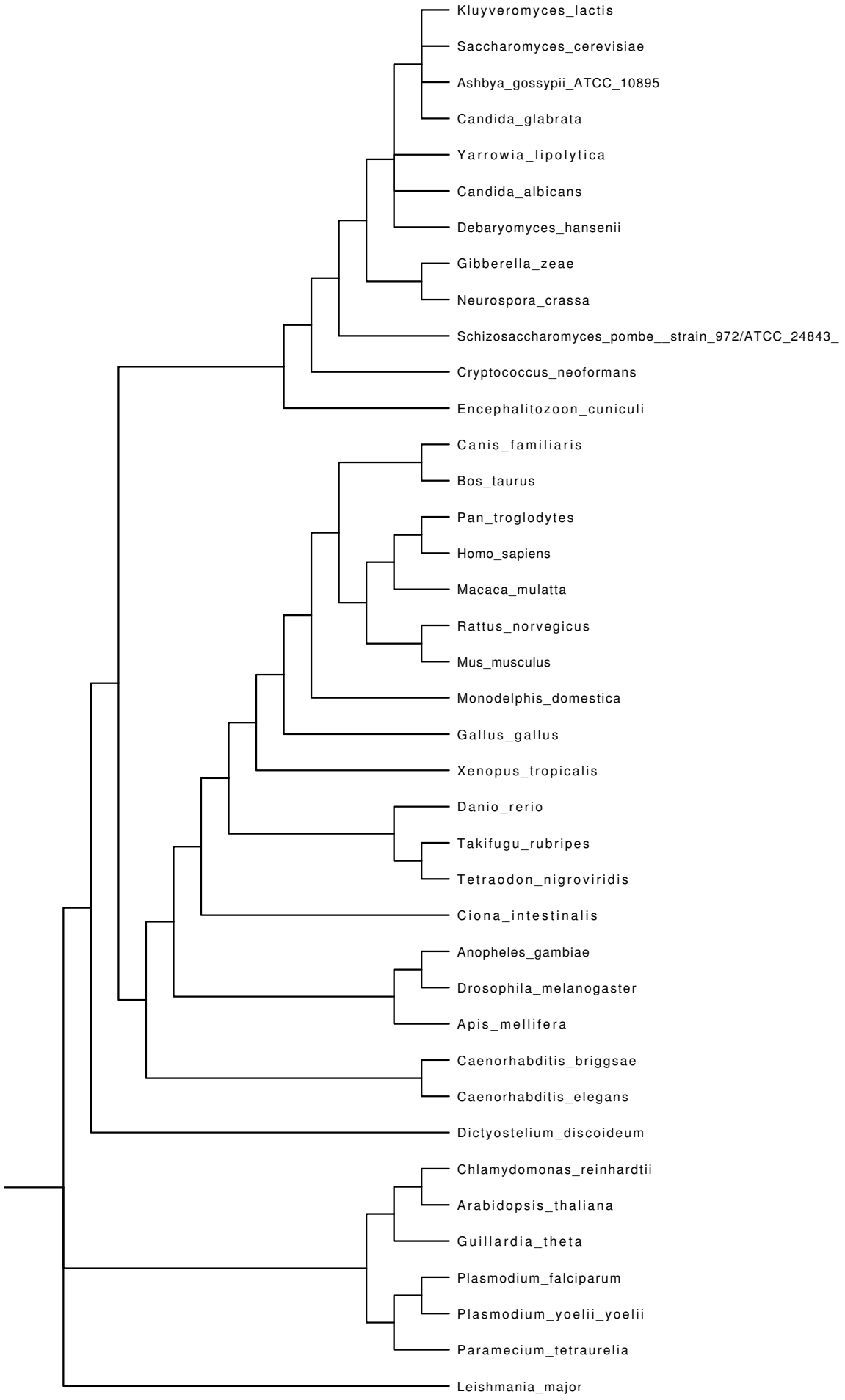



**Figure 7.2:** Species tree of 39 Eukaryotes used for the validation against PhylomeDB.

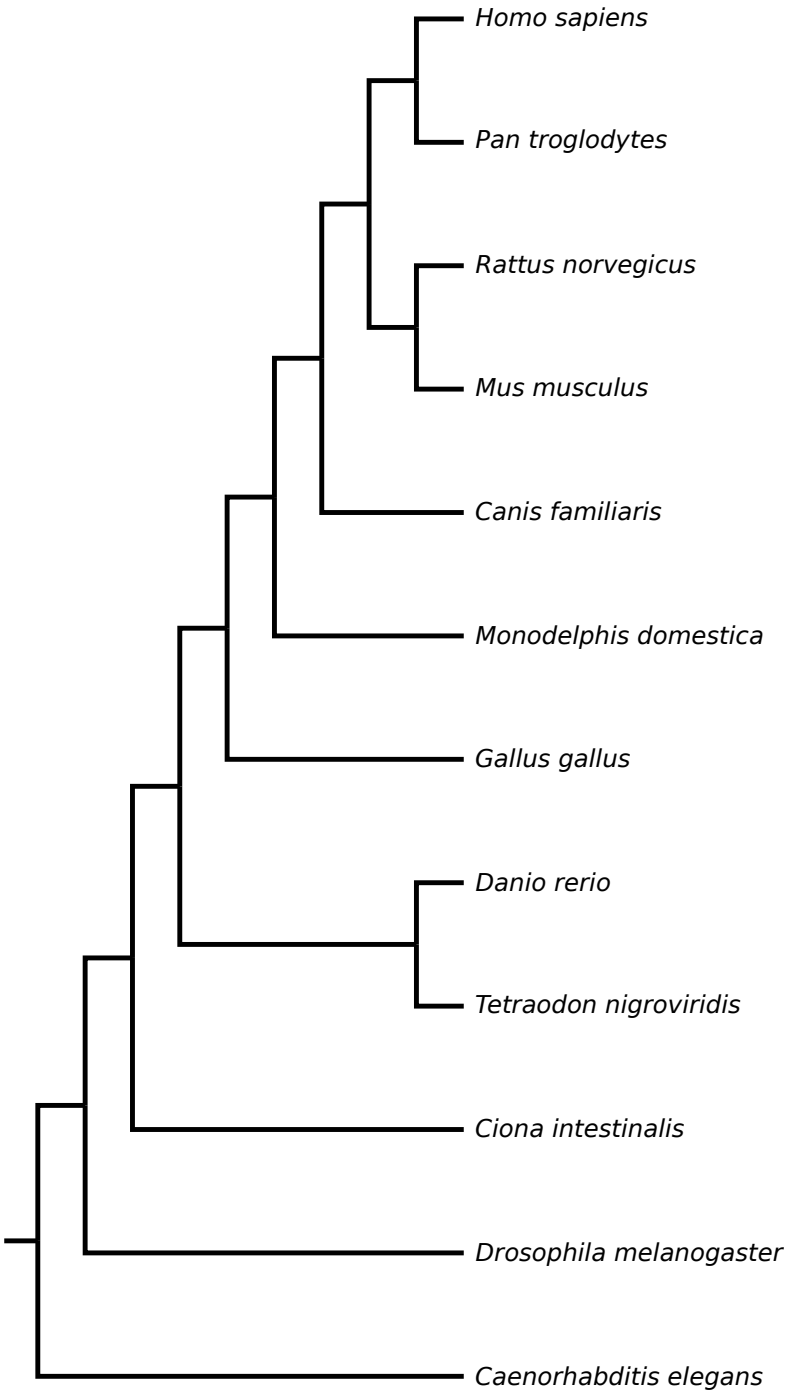



**Figure 7.3:** Species tree of 12 species used for the validation against the set of 70 manually curated families.

# Bibliography


Altenhoff, A. M. and Dessimoz, C. (2009). Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS computational biology*, 5(1):e1000262.

Altenhoff, A. M., Gil, M., Gonnet, G. H., and Dessimoz, C. (2013). Inferring hierarchical orthologous groups from orthologous gene pairs. *PloS one*, 8(1):e53786.

Altenhoff, A. M., Schneider, A., Gonnet, G. H., and Dessimoz, C. (2011). OMA 2011: orthology inference among 1000 complete genomes. *Nucleic acids research*, 39(Database issue):D289–94.

Chen, F., Mackey, A. J., Stoeckert, C. J., and Roos, D. S. (2006). OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic acids research*, 34(Database issue):D363–8.

Chen, F., Mackey, A. J., Vermunt, J. K., and Roos, D. S. (2007). Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PloS one*, 2(4):e383.

Coletta, A., Pinney, J. W., Solís, D. Y. W., Marsh, J., Pettifer, S. R., and Attwood, T. K. (2010). Low-complexity regions within protein sequences have position-dependent roles. *BMC systems biology*, 4:43.

Dalquen, D. a., Altenhoff, A. M., Gonnet, G. H., and Dessimoz, C. (2013). The Impact of Gene Duplication, Insertion, Deletion, Lateral Gene Transfer and Sequencing Error on Orthology Inference: A Simulation Study. *PLoS ONE*, 8(2):e56925.

Dessimoz, C., Cannarozzi, G., Gil, M., Margadant, D., Roth, A., Schneider, A., and Gonnet, G. H. (2005). OMA , A Comprehensive , Automated Project for the Identification of Orthologs from Complete Genome Data : Introduction and First Achievements. pages 61–72.

Diekmann, Y., Seixas, E., Gouw, M., Tavares-Cadete, F., Seabra, M. C., and Pereira-Leal, J. B. (2011). Thousands of rab GTPases for the cell biologist. *PLoS computational biology*, 7(10):e1002217.

Ereshefsky, M. (2012). Homology thinking. *Biology & Philosophy*, 27(3):381–400.

Fitch, W. M. (1970). Distinguishing Homologous from Analogous proteins. *Systematic Zoology*, 19(2):99–113.

Fitch, W. M. (2000). Homology. *Trends in genetics : TIG*, 16(5).

Forslund, K., Pekkari, I., and Sonnhammer, E. L. L. (2011). Domain architecture conservation in orthologs. *BMC bioinformatics*, 12(1):326.

Gabaldón, T. (2008). Large-scale assignment of orthology: back to phylogenetics? *Genome biology*, 9(10):235.

Gough, J., Karplus, K., Hughey, R., and Chothia, C. (2001). Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *Journal of molecular biology*, 313(4):903–19.

Huerta-Cepas, J., Bueno, A., Dopazo, J., and Gabaldón, T. (2008). PhylomeDB: a database for genome-wide collections of gene phylogenies. *Nucleic acids research*, 36(Database issue):D491–6.

Huerta-Cepas, J., Capella-Gutierrez, S., Pryszcz, L. P., Denisov, I., Kormes, D., Marcet-Houben, M., and Gabaldón, T. (2011a). PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. *Nucleic acids research*, 39(Database issue):D556–60.

Huerta-Cepas, J., Dopazo, H., Dopazo, J., and Gabaldón, T. (2007). The human phylome. *Genome biology*, 8(6):R109.

Huerta-Cepas, J., Dopazo, J., Huynen, M. a., and Gabaldón, T. (2011b). Evidence for short-time divergence and long-time conservation of tissue-specific expression after gene duplication. *Briefings in bioinformatics*, 12(5):442–8.

Jensen, L. J., Julien, P., Kuhn, M., von Mering, C., Muller, J., Doerks, T., and Bork, P. (2008). eggNOG: automated construction and annotation of orthologous groups of genes. *Nucleic acids research*, 36(Database issue):D250–4.

Koonin, E. V. (2005). Orthologs, paralogs, and evolutionary genomics. *Annual review of genetics*, 39:309–38.

Kristensen, D. M., Wolf, Y. I., Mushegian, A. R., and Koonin, E. V. (2011). Computational methods for Gene Orthology inference. *Briefings in bioinformatics*, 12(5):379–91.

Kriventseva, E. V., Rahman, N., Espinosa, O., and Zdobnov, E. M. (2008). OrthoDB: the hierarchical catalog of eukaryotic orthologs. *Nucleic acids research*, 36(Database issue):D271–5.

Kuzniar, A., van Ham, R. C. H. J., Pongor, S., and Leunissen, J. a. M. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends in genetics : TIG*, 24(11):539–51.

Li, H., Coghlan, A., Ruan, J., Coin, L. J., Hériché, J.-K., Osmotherly, L., Li, R., Liu, T., Zhang, Z., Bolund, L., Wong, G. K.-S., Zheng, W., Dehal, P., Wang, J., and Durbin, R. (2006). TreeFam: a curated database of phylogenetic trees of animal gene families. *Nucleic acids research*, 34(Database issue):D572–80.

Li, L., Stoeckert, C. J., and Roos, D. S. (2003). OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–89.

McDonnell, a. V., Jiang, T., Keating, a. E., and Berger, B. (2006). Paircoil2: improved prediction of coiled coils from sequence. *Bioinformatics (Oxford, England)*, 22(3):356–8.

Nehrt, N. L., Clark, W. T., Radivojac, P., and Hahn, M. W. (2011). Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS computational biology*, 7(6):e1002073.

Ostlund, G., Schmitt, T., Forslund, K., Köstler, T., Messina, D. N., Roopra, S., Frings, O., and Sonnhammer, E. L. L. (2010). InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. *Nucleic acids research*, 38(Database issue):D196–203.

Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D., and Maltsev, N. (1999). The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6):2896–901.

Peterson, M. E., Chen, F., Saven, J. G., Roos, D. S., Babbitt, P. C., and Sali, A. (2009). Evolutionary constraints on structural similarity in orthologs and paralogs. *Protein science : a publication of the Protein Society*, 18(6):1306–15.

Powell, S., Szklarczyk, D., Trachana, K., Roth, A., Kuhn, M., Muller, J., Arnold, R., Rattei, T., Letunic, I., Doerks, T., Jensen, L. J., von Mering, C., and Bork, P. (2012). eggNOG v3.0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. *Nucleic acids research*, 40(Database issue):D284–9.

Rackham, O. J. L., Madera, M., Armstrong, C. T., Vincent, T. L., Woolfson, D. N., and Gough, J. (2010). The evolution and structure prediction of coiled coils across all genomes. *Journal of molecular biology*, 403(3):480–93.

Remm, M., Storm, C. E. V., and Sonnhammer, E. L. L. (2001). Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons. pages 1041–1052.

Ruan, J., Li, H., Chen, Z., Coghlan, A., Coin, L. J. M., Guo, Y., Hériché, J.-K., Hu, Y., Kristiansen, K., Li, R., Liu, T., Moses, A., Qin, J., Vang, S. r., Vilella, A. J., Ureta-Vidal, A., Bolund, L., Wang, J., and Durbin, R. (2008). TreeFam: 2008 Update. *Nucleic acids research*, 36(Database issue):D735–40.

Scannell, D. R., Byrne, K. P., Gordon, J. L., Wong, S., and Wolfe, K. H. (2006). Multiple rounds of speciation associated with reciprocal gene loss in polyploid yeasts. *Nature*, 440(7082):341–5.

Sonnhammer, E. L. and Koonin, E. V. (2002). Orthology, paralogy and proposed classification for paralog subtypes. *TRENDS in Genetics*, 18(12):619–620.

Stevcic, Z. (1978). On Phylogenetic Reconstruction. 27(2):227.

Storm, C. E. V. and Sonnhammer, E. L. L. (2002). Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. 18(1):92–99.

Studer, R. a. and Robinson-Rechavi, M. (2009). How confident can we be that orthologs are similar, but paralogs differ? *Trends in genetics : TIG*, 25(5):210–6.

Tatusov, R. L. (1997). A Genomic Perspective on Protein Families. *Science*, 278(5338):631–637.

Trachana, K., Larsson, T. a., Powell, S., Chen, W.-H., Doerks, T., Muller, J., and Bork, P. (2011a). Orthology prediction methods: a quality assessment using curated protein families. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 33(10):769–80.

Trachana, K., Larsson, T. a., Powell, S., Chen, W.-H., Doerks, T., Muller, J., and Bork, P. (2011b). Orthology prediction methods: a quality assessment using curated protein families. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 33(10):769–80.

van der Heijden, R. T. J. M., Snel, B., van Noort, V., and Huynen, M. a. (2007). Orthology prediction at scalable resolution by phylogenetic tree analysis. *BMC bioinformatics*, 8:83.

Vilella, A. J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R., and Birney, E. (2009). EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome research*, 19(2):327–335.

Wall, D. P., Fraser, H. B., and Hirsh, a. E. (2003). Detecting putative orthologs. *Bioinformatics*, 19(13):1710–1711.